

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

PCT

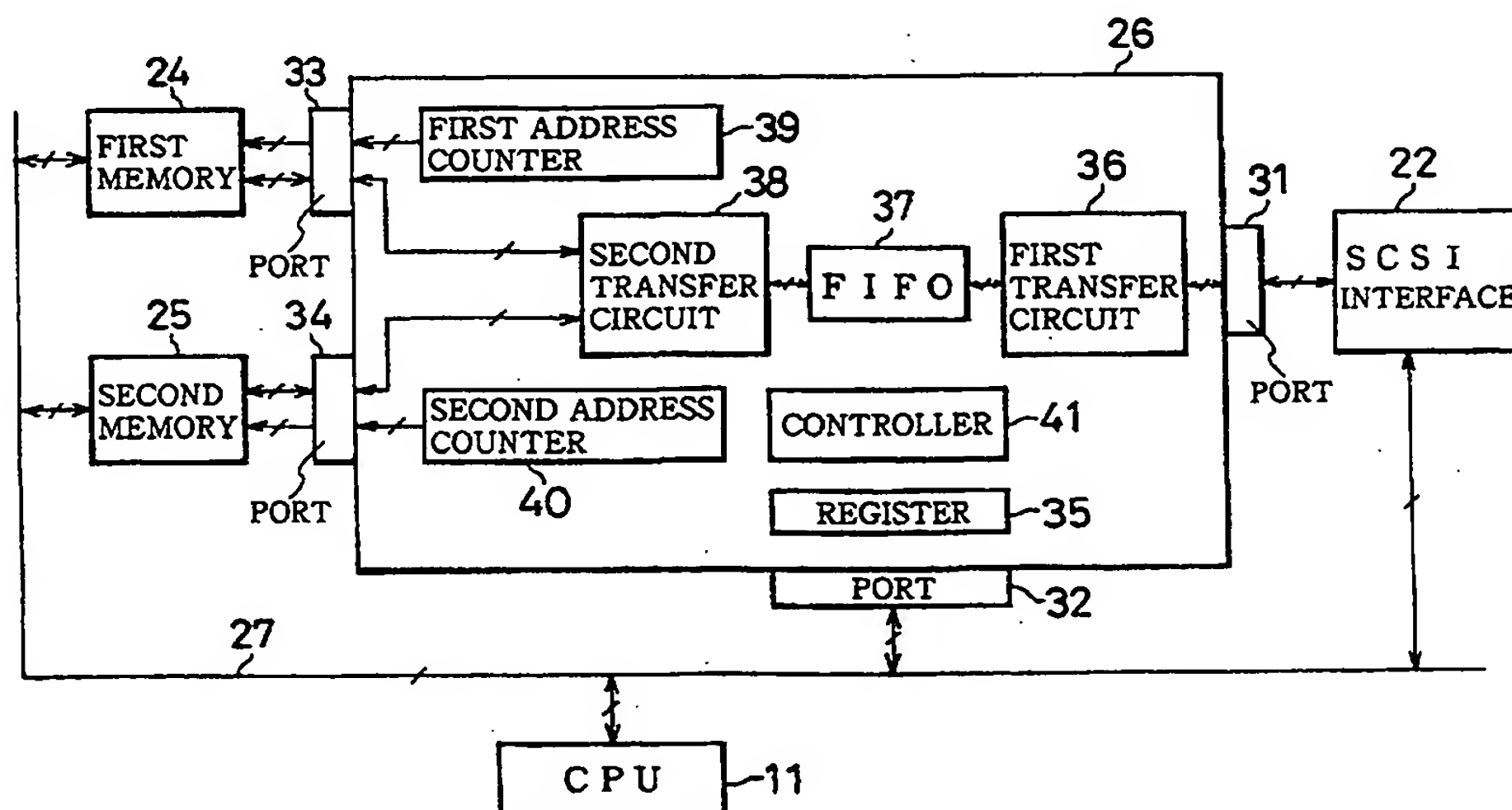
WORLD INTELLECTUAL PROPERTY ORGANIZATION  
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> : <b>G06F 3/06</b>	<b>A1</b>	(11) International Publication Number: <b>WO 96/35986</b> (43) International Publication Date: 14 November 1996 (14.11.96)
(21) International Application Number: PCT/JP96/01210 (22) International Filing Date: 7 May 1996 (07.05.96) (30) Priority Data: 7/134683 9 May 1995 (09.05.95) JP (71) Applicant: SOFMAP FUTURE DESIGN CO., LTD. [JP/JP]; 3-14-10, Sotokanda, Chiyoda-ku, Tokyo 101 (JP). (72) Inventors: SHOJI, Wataru; Sofmap Future Design Co., Ltd., 3-14-10, Sotokanda, Chiyoda-ku, Tokyo 101 (JP). TABUCHI, Daisuke; Sofmap Future Design Co., Ltd., 3-14-10, Sotokanda, Chiyoda-ku, Tokyo 101 (JP). NAKAJIMA, Ichiro; Sofmap Future Design Co., Ltd., 3-14-10, Sotokanda, Chiyoda-ku, Tokyo 101 (JP). (74) Agent: KIMURA, Mitsuru; Kanda-Sakata Building, 6th floor, 2-5-1, Kandanishiki-cho, Chiyoda-ku, Tokyo 101 (JP).		(81) Designated States: AU, BR, CA, CN, KR, MX, PL, RU, SG, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).  Published With international search report.

(54) Title: INTERFACE CIRCUIT AND DATA PROCESSING APPARATUS AND METHOD



(57) Abstract

A data to be processed by CPU (11) is sequentially read from an external storage device (21), and transferred to first and second memories (24, 25) alternatively. CPU (11) accesses the one of the first and second memories (24, 25) alternatively, and processes the data previously read.

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(51) Int. Cl. 6		(11) 공개번호	특 1999-008415
G06F 3 /06		(43) 공개일자	1999년01월25일
(21) 출원번호	특 1997-707942		
(22) 출원일자	1997년11월07일		
번역문제출일자	1997년11월07일		
(86) 국제출원번호	PCT/JP 96/001210	(87) 국제공개번호	W0 96/035986
(86) 국제출원출원일자	1996년05월07일	(87) 국제공개일자	1996년11월14일
(81) 지정국	EP 유럽특허 : 오스트리아 벨기에 스위스 독일 덴마크 스페인 프랑스 영국 그리스 아일랜드 이탈리아 룩셈부르크 모나코 네덜란드 포르투갈 스웨덴 핀란드 국내특허 : 오스트레일리아 브라질 캐나다 중국 대한민국 멕시코 폴란드 러시아 싱가포르		
(30) 우선권주장	95-134683 1995년05월09일 일본(JP)		
(71) 출원인	소프맵퓨처디자인가부시키가이샤 스즈키게이		
(72) 발명자	일본국 도쿄도 지요다쿠 소토칸다 3-14-10 쇼지와타루		
	일본국 도쿄도 지요다쿠 소토칸다 3-14-10 소프맵 퓨처 디자인 가부시키가이샤 내 다부치다이스케		
	일본국 도쿄도 지요다쿠 소토칸다 3-14-10 소프맵 퓨처 디자인 가부시키가이샤 내 나카지마이치로		
(74) 대리인	일본국 도쿄도 지요다쿠 소토칸다 3-14-10 소프맵 퓨처 디자인 가부시키가이샤 내 손은진		

심사청구 : 있음

(54) 데이터 전송시스템 및 그 방법

요약

CPU(11)로 처리되는 데이터는 외부기억장치(21)에서 순차적으로 판독되고, 제1 및 제2메모리(24,25)에 양자택일로 전송된다. CPU(11)는 제1 및 제2메모리(24,25)중의 하나를 양자택일로 액세스하고 이전에 판독된 데이터를 처리한다.

대표도

도2

## 명세서

## 기술분야

본 발명은 외부기억장치와 CPU(중앙처리장치) 사이에서 빠른 데이터 전송이 가능한 인터페이스 회로에 관한 것이다.

또한 본 발명은 외부기억장치를 고속으로 액세스할 수 있는 데이터 처리장치 및 그 방법에 관한 것이다.

## 배경기술

보통의 데이터 처리장치는 하드디스크나 CD-ROM과 같이 외부기억장치를 갖추며 내부메모리와 외부기억장치 사이에 데이터 전송을 실행하여 데이터 처리를 수행한다.

그러나 외부기억장치의 액세스시간이 CPU의 동작속도에 비해 매우 느려서 데이터처리의 작업량이 감소한다. 특히 방대한 양의 비트맵 이미지 데이터 등을 처리하는 애플리케이션은 외부기억장치를 이용하여 데이터를 전송하는 시간이 걸리는데 이것은 오퍼레이터의 생각을 간섭한다.

## 발명의 요약

따라서 본 발명의 목적은 컴퓨터의 외부기억장치와 내부메모리 사이에서 데이터 전송을 고속으로 하는 방법 및 장치를 제공하는 것이다. 고속 데이터 전송장치는 제1메모리, 제2메모리, 외부기억장치와 데이터를 통신하기 위한 제1회로, 내부메모리와 데이터를 통신하기 위한 제2메모리를 포함한다. 또한 이 장치는 제1 및 2메모리중의 하나에 제1회로를 양자택일로 연결하고 제1 및 2메모리중의 남은 하나에 제2회로를 양자택일로 연결하는 데이터 스위칭회로를 포함한다. 제1회로가 제1 혹은 제2메모리에 연결될 때 외부기억장치와 제1 혹은 제2메모리 사이에 데이터가 전송된다. 제2회로가 제1 혹은 제2메모리에 연결될 때 제1 혹은 제2메모리와 내부메모리 사이에 데이터가 전송된다.

위의 데이터 전송장치의 장점 중에 하나는 제1 및 제2회로가 다른 데이터 전송동작을 동시에 수행한다는 것이다. 예를들어 외부기억장치는 제1메모리에 데이터를 전송할 수 있는 반면에 내부메모리는 제2메모리로부터 데이터를 전송할 수 있다. 제1메모리가 가득찰 때 제2메모리는 비어있다. 그 다음에 외부기억장치는 제2메모리로 스위치될 수 있다. 따라서 외부기억장치는 제2메모리에 데이터를 계속하여 전송하는 반면에 내부메모리는 제1메모리로부터 데이터를 계속하여 전송한다. 그 결과 외부기억장치와 내부메모리 사이에 데이터가 거의 계속하여 전송될 수 있다.

바람직한 실시예에서 제1메모리, 제2메모리, 내부메모리는 반도체 장치이다. 외부기억장치는 기계적 장치이므로 데이터 전송에 필요한 시간이 길다. 따라서 내부메모리로부터 혹은 내부메모리로의 데이터 전송은 외부기억장치가 여전히 동작하는 동안 완료될 것이다. 외부기억장치는 본 발명의 데이터 전송장치에 의해 제어되기 때문에 컴퓨터의 CPU는 정상적으로 대기하는 시간일 것이다. 본 발명의 한 실시예는 CPU 유휴시간을 이용하는 방법이다.

데이터 전송장치는 데이터 스위칭에 관련된 상태신호를 발생시키는 수단을 추가로 포함한다. 상태신호에 응답하는 소프트웨어 드라이버는 CPU가 데이터 전송에 필요하지 않는 시간동안 다른 소프트웨어 프로그램이 실행되게 한다. 그 결과로서 CPU는 대기하는 유휴가 없게 될 것이다..

독창적인 데이터 전송장치 및 독창적 소프트웨어 드라이버는 외부기억장치를 액세스하면서 프로그램 실행속도를 빠르게 할 때 공동으로 동작하기 때문에 독창적 소프트웨어는 종래의 드라이버(IBM 퍼스널 컴퓨터사의 BIOS 하드디스크장치 서비스 루틴등)를 대체하는데 이용되어야 한다.

다른 특징과 장점은 첨부된 도면과 함께 다음의 발명의 상세한 설명으로부터 알 수 있다.

#### 도면의 간단한 설명

도 1은 본 발명의 제1실시예에 따른 데이터 처리장치의 구조를 도시한 블록도이다.

도 2는 도 1의 전송제어장치의 구조를 도시한 블록도이다.

도 3은 하드디스크에서 데이터를 판독하는 시간과 판독된 데이터를 처리하는 시간 사이의 관계를 도시한 타이밍 차트이다.

도 4는 맵데이터를 구성하는 복수개의 블록을 설명하는데 이용된 도이다.

도 5는 맵데이터로 정의된 맵에 열린 윈도우를 도시한 도이다.

도 6은 맵상의 윈도우 이동을 도시한 도이다.

도 7은 윈도우에서 이미지의 시프팅을 도시한 도이다.

도 8a 및 8b는 윈도우에서 이미지의 시프팅을 도시한 도이다.

도 9는 데이터 처리장치의 변형을 도시한 블록도이다.

도 10은 본 발명의 제2실시예에 따른 데이터 처리장치와 통합되는 컴퓨터 시스템의 블록도이다.

도 11은 데이터 전송장치의 동작을 도시한 흐름도이다.

도 12는 데이터 전송장치의 또다른 동작모드를 도시한 흐름도이다.

도 13은 데이터 전송장치의 동작중에 다양한 사이클 사이의 상대적 타이밍을 도시한다.

도 14는 윈도우즈 기본환경에서 동작하는 본 발명에 따라 데이터전송 인터페이스와 소프트웨어 드라이버를 포함하는 컴퓨터 시스템의 블록도이다.

도 15는 도 14에 도시한 시스템 동작을 도시한 흐름도이다.

#### 도면의 주요부분에 대한 부호의 설명

11, 102: CPU	22: SCSI 인터페이스	24: 제1메모리
25: 제2메모리	26: 전송제어장치	27: 제2버스
31: SCSI 포트	32: CPU 포트	33: 제1포트
34: 제2포트	35: 레지스터	36: 제1전송회로
37: FIFO	38: 제2전송회로	39, 136: 제1주소카운터
40, 138: 제2주소카운터	41, 122: 제어장치	104: 내부메모리
112: 외부기억장치	134: 스위칭회로	152: 상태신호회로

## 제1실시예

이하, 본 발명의 제1실시예에 따른 데이터 처리장치 및 인터페이스 회로에 대해 설명한다.

도 1은 제1실시예에 따른 데이터 처리장치의 구조를 도시한다. 도시한 바와 같이 이 데이터 처리장치는 CPU(중앙처리장치)(11), ROM(13), 비디오 RAM(VRAM)(14), 그리고 VRAM의 내용을 표시하는 표시장치(15)를 포함한다. ROM(13) 및 VRAM(14)은 제1버스(12)에 의해 CPU(11)에 접속된다. CPU(11)는 데이터처리 및 시스템의 일반적 제어를 수행한다. 예를들어 제1버스(12)는 EISA 표준에 부합하도록 설계되고 데이터버스, 주소버스, 제어버스를 포함한다. ROM(13)은 CPU(11)에 의해 실행되는 제어프로그램, 고정 데이터 등을 기억하고 있다. VRAM(14)은 표시데이터를 정의하는 비트맵 데이터를 기억한다. 표시장치(15)는 CRT나 액정표시장치 등으로 구성된다.

이 데이터 처리장치는 하드디스크나 CD-ROM 등의 외부기억장치(보조기억장치)(21)를 추가로 포함한다. 이 실시예에서 외부기억장치(21)는 하드디스크이다. SCSI(소형 컴퓨터 시스템 인터페이스)표준에 부합하는 하드디스크(21)는 병렬회선을 거쳐 SCSI 인터페이스(22)에 접속된다. 이 실시예에 따라서 래스터형(비트맵형)이미지 데이터는 하드디스크(21)에 직렬로 기억되어 액세스가 빨리된다. SCSI 인터페이스(22)는 전송인터페이스(23)에 접속된다. 전송인터페이스(23)는 CPU(11)에 의해 액세스되는 제1 및 제2메모리(24 및 25)와 전송제어장치(26)를 구비한다.

각각의 제1 및 제2메모리(24,25)는 2중포트 SRAM 메모리같은 2중포트 패스트 메모리로 구성된다. 제1 및 제2메모리(24,25)의 제1포트는 제2버스(27)에 의해 CPU(11)에 접속된다. CPU(11)는 제1포트를 통해 이들 메모리(24,25)로부터 데이터를 판독하거나 이들 메모리에 데이터를 기록한다. 제2버스는 예를들면 EISA표준과 부합한다.

전송제어장치(26)는 제1 및 제2메모리(24,25)의 제1포트, SCSI 인터페이스(22), 및 제2버스(27)에 접속되고 CPU(11)의 제어하에 하드디스크(21)와 제1 및 제2메모리(24,25) 사이의 데이터 전송을 제어한다.

도 2에 도시한 바와 같이 전송제어장치(26)는 SCSI 포트(31), CPU 포트(32), 제1포트(33), 제2포트(33), 레지스터(35), 제1전송회로(36), FIFO(First In First Out Memory)(37), 제2전송회로(38), 제1주소카운터(39), 제2주소카운터(40), 제어장치(41)를 구비한다.

SCSI 포트(31)는 배타적 회선을 거쳐 SCSI 인터페이스(22)에 접속된다. CPU 포트(32)는 제2버스(27)에 접속된다. 제1포트(33)는 배타적 라인을 거쳐 제1메모리(24)의 제2포트에 접속된다. 제2포트(34)는 배타적 회선을 거쳐 제2메모리(25)의 제2포트에 접속된다.

플래그, 기록이나 판독을 지시하는 데이터, CPU(11)에 의해 설정된 헤드주소등은 레지스터(35)에 설정된다.

제1전송회로(36)는 FIFO(37)와 SCSI 인터페이스(22)(하드디스크(21))사이에 데이터를 전송한다. 제2전송회로(38)는 FIFO(37)와 제1메모리(24)나 제2메모리(25) 사이에 데이터를 전송한다. 제1전송회로(36) 및 제2전송회로(38)는 예를들면 DMA(직접 메모리 접근)회로로 구성된다. 제1전송회로는 예를들면 20MHz의 동작클럭에 따라 동작하고 제2전송회로(38)는 예를들면 40MHz의 동작클럭에 따라 동작한다.

FIFO(37)는 제1전송회로(36)의 데이터 전송속도와 제2전송회로(38)의 데이터 전송속도 사이의 차이를 완충하여, 제1 및 제2전송회로(36,38) 사이의 데이터 교환을 위해 이들(36,38) 사이에 개입한다.

제1주소카운터(39)는 제1메모리(24)의 주소를 카운트한다. 제2주소카운터(40)는 제2메모리(25)의 주소를 카운트한다.

제어장치(41)는 전송제어장치(26) 내부회로의 각 부의 동작을 제어한다.

이하 전송 인터페이스(23)의 기본동작에 대해 설명한다.

맨 먼저, CPU(11)는 데이터 판독을 지시하는 명령, 판독되는 데이터가 기억되어 있는 하드디스크(21)에 있는 영역의 헤드(시작)주소, 전송요구신호(DMA 요구)를 제2버스(27)를 거쳐 전송제어장치(26)에 공급한다. 전송요구신호(DMA요구)는 제1

및 제2메모리(24,25)중의 하나가 우선 기억되어야 하는 것을 지시하는 데이터를 포함한다.

제어장치(41)는 연관 레지스터(35)에 있는 전송요구신호에 포함된 데이터, 플래그 등을 설정하고 연관된 동작모드에서 각 부를 설정한다. 제어장치(41)는 제1 및 제2주소카운터(39 및 40)를 초기화한다. 또한 전송요구신호는 SCSI 인터페이스(22)에 공급된다.

설정이 완료될 때 제어장치(41)는 CPU(11)에 긍정응답신호(ACK)를 낸다. 이 신호(ACK)는 데이터 전송동작으로부터 CPU(11)를 자유롭게 하여 CPU(11)가 다른 처리를 실행하게끔 한다.

CPU(11)에서의 전송요구에 응하여 SCSI 인터페이스(22)는 하드디스크에 데이터를 판독하라고 명령한다. 이 명령에 응하여 하드디스크(21)는 헤드주소에서 데이터 판독을 시작하고 배타적 회선을 거쳐 SCSI 인터페이스(22)에 데이터를 보낸다.

SCSI 인터페이스(22)로부터 데이터가 공급될때마다 제1전송회로(36)는 FIFO(37)에 데이터를 보낸다. 제2전송회로(38)는 FIFO(37)에 입력된 데이터를 순차적으로 포착하고 이것을 CPU(11)에 의해 지정된 메모리, 예를들면 제1메모리(24)에 보낸다. 제1메모리(24)는 공급된 데이터를 제1주소카운터(39)에 의해 지정된 위치에 기억시킨다. 제2전송회로(38)가 데이터를 한 번 전송할때마다 제1주소카운터(39)는 카운트값(주소값)을 갱신한다. 갱신값은 단일 전송처리에서 전송되는 데이터의 바이트 수와 대응한다.

그런 다음에 데이터는 하드디스크(21)에서 순차적으로 판독되고 제1메모리(24)에 기억된다.

데이터 전송이 진행되고 소정의 양, 예를들면 64킬로바이트의 데이터 전송이 완료됨에 따라 전송 제어장치(26)는 전송종료 인터럽트 신호(INT)를 제2버스(27)를 거쳐 CPU(11)에 보낸다. 인터럽트신호(INT)에 응답하는 인터럽트 처리에서 CPU(11)는 데이터가 다음에 판독되어야 하는 영역의 헤드위치같은 정보를 포함하는 전송요구신호를 출력하고 전송제어장치(26)를 재가동시킨다. 이런 재가동으로 인해 데이터가 다시 전송되게 된다.

데이터전송이 진행되고 제1메모리(24)가 전송된 데이터로 채워짐에 따라 전송 제어장치(26)는 인터럽트신호(INT)를 CPU(11)에 공급한다. 인터럽트신호(INT)에 응답하는 인터럽트 처리에서 CPU(11)는 목표 메모리로서 제2메모리(25)를 지시하는 전송요구신호를 전송제어장치(26)에 출력한다. 제어장치(41)는 제2주소카운터(40)의 카운트값을 초기화한다. 그런 다음 하드디스크(21)에서 판독된 데이터가 제2메모리(25)에 순차적으로 전송된다.

데이터가 제2메모리(25)에 기억되는 동안 CPU(11)는 제1메모리(24)로부터 데이터를 판독하고 이것을 처리한다.

제2메모리(25)로의 데이터 전송과 제1메모리(24)에 기억된 데이터의 처리가 모두 완료될 때 CPU(11)는 제1메모리(24)로의 데이터전송과 판독위치 등을 지시하는 전송요구신호를 전송제어장치(26)에 보낸다. 이 요구신호에 따라 제어장치(41)는 제1주소카운터(39)를 초기화한다. 그 결과 하드디스크(21)에서 판독된 데이터는 CPU(11)가 제2메모리(25)에 기억된 데이터를 처리하는 시간동안 제1메모리(24)에 순차적으로 기억된다.

그 다음은 같은 동작이 반복된다. 좀더 자세히 설명하면 CPU(11)는 제1 및 제2메모리(24,25)에 기억된 데이터를 양자택일로 반복적으로 처리한다.

그리고 전송제어장치(26)는 다음에 처리되는 데이터를 CPU(11)가 액세스하고 있는 메모리와는 다른 메모리에 전송한다.

전송되는 데이터의 마지막 블록의 전송을 지시하는 인터럽트신호(INT)가 CPU(11)에 보내질 때 데이터 전송이 종료된다.

위에 기술한 동작의 처리타이밍은 도 3에 도시되어 있다. 이 도에서 명백한 바와 같이 하드디스크(21)에서 데이터의 처리요구가 발생한 후에 전송된 데이터를 대기하는 CPU(11)의 기간은 제1메모리(24)가 먼저 채워지는 기간뿐이다.

그런 다음에, 처리되는 데이터는 연관 메모리에 미리 준비되어 있다. 그러므로 데이터 전송속도에 제한되지 않고 빠른 데이터처리가 가능하다.



CPU(11)의 처리가 처리되는 데이터 전송이 끝나기 전에 완료될 때, CPU(11)는 다른 처리를 실행하고 전송의 종료를 지시하는 인터럽트 신호(INT)에 따라 데이터 처리로 돌아간다. 다음에 처리되는 데이터 전송이 CPU(11)의 처리가 끝나기 전에 완료될 때 전송 제어장치(26)는 CPU(11)로부터 다음 명령을 대기한다.

이하 CPU(11)에 의해 처리된 방대한 양의 데이터가 하드디스크(21)에 순차적으로 전송되는 경우에 대해 설명한다.

우선 CPU(11)는 데이터 처리를 실행하고 제1 및 제2메모리(24,25)에 데이터를 기억시킨다. 소정 양의 데이터가 생성될 때 CPU(11)는 데이터기록을 지시하는 명령과 전송요구신호를 제2버스(27)를 거쳐 전송제어장치(26)에 보낸다. 전송요구신호는 제1메모리(24)에 있는 데이터가 먼저 전송되어야 한다는 것과, 기록영역의 헤드주소 등을 지시한다. 제어장치(41)는 연관모드로 각 부를 설정한다. CPU(11)는 SCSI 인터페이스(22)에 데이터 기록모드와 하드디스크(21)에 있는 목표기록영역의 헤드주소를 알린다.

설정이 완료될 때 제어장치(41)는 CPU(11)에 긍정응답신호(ACK)를 보낸다. 이 신호(ACK)로 CPU(11)는 데이터 전송동작으로부터 자유롭게 되어 데이터 처리를 재개하고 제2메모리(25)에 데이터 처리에서 생기는 데이터를 기록한다.

전송요구신호에 따라 제2전송회로(38)는 주소카운터(39)에 의해 지정된 제1메모리(24)에 있는 위치에서 데이터를 순차적으로 판독하고 FIFO(37)에 데이터를 기억시킨다. 제1전송회로(36)는 FIFO(37)로부터 데이터를 획득하고 이 데이터를 SCSI 인터페이스(22)에 공급한다. SCSI 인터페이스(22)는 지정된 헤드주소로부터 공급데이터를 순차적으로 기록한다. 그 다음에는 같은 동작이 반복된다. 그 결과 데이터 한 블록이 전송될때마다 인터럽트신호(INT)가 CPU(11)에 보내진다. 인터럽트신호(INT)에 응답하여 CPU(11)는 데이터 전송을 다시 요구한다.

제1메모리(24)에서 기억된 데이터의 전송이 완료되고 제2메모리(25)에 데이터 기록이 완료될 때 CPU(11)는 전송메모리를 제2메모리로 스위치하고 기록메모리를 제1메모리로 스위치한다. 제2전송회로(38)는 제2주소카운터(40)에 의해 지시된위치에서 데이터를 순차적으로 판독하고 이 데이터를 FIFO(37)에 입력한다. 그러는 동안 제1전송회로(36)는 FIFO(37)로부터 데이터를 순차적으로 판독하고 이것을 SCSI 인터페이스(22)에 공급한다. SCSI 인터페이스(22)는 수신된 데이터를 하드디스크(21)에 공급하고 여기에 기록한다.

그런 후에 같은 동작이 반복된다.

데이터 기록시, CPU(11)가 하나의 메모리(24 혹은 25)를 액세스하여 데이터를 준비하는 동안 다른 메모리(25 혹은 24)에 이미 준비된 데이터가 판독되어 하드디스크(21)에 전송된다. 그러므로 CPU(11)는 하드디스크(21)와의 데이터 교환에 필요한 시간으로 방해받지 않고 데이터 처리를 실행할 수 있다. 따라서 CPU(11)는 이미지 데이터 처리와 같은 방대한 양의 데이터를 빠른 속도로 처리할 수 있다.

디스크 고속처리(caching)와 달라서, 이런 데이터 전송은 데이터의 히트(hit) 혹은 언히트(unhit)의 결정, 또는 언히트한 경우에 데이터의 대체 등을 결정할 필요가 없다. 이에따라 제어시스템의 구조를 간소화하고 동작속도를 향상시킬 수 있다.

이하 데이터 처리장치의 일반동작에 대하여 윈도우가 맵같은 이미지상에 열리고 이 윈도우가 맵 상에서 움직이는 동안 표시되는 경우와 관련하여 자세히 설명한다.

이 실시예에서 래스터형(비트맵형)맵 데이터가 하드디스크(21)에 순차적으로 기억된다는 것을 주목해야 한다. 맵데이터는 도 4에 도시한 바와 같이, 512 X 512비트의 매트릭스로 각각 구성되는 복수개의 블록으로 나뉘어진다. 데이터의 각 피스(piece)는 8비트의 깊이(비트 수)를 갖는다.

ROM(13)에 기억된 제어프로그램에 따라 CPU(11)는 맵데이터로 정의된 큰 가상 맵(51)상에 윈도우(52)를 형성시키고 윈도우(52)를 통해 맵(51)의 일부를 표시한다. 윈도우(52)와 이 윈도우(52)상의 맵(51)은 비트맵형태로 나타나 표시장치(15)에 표시된다.



이런 상황에서 사용자는 오른쪽 상단으로 화면이동을 명령하고(윈도우는 오른쪽 상단으로 이동하고 맵은 왼쪽 하단으로 이동도록 나타난다), CPU(11)는 도 6에 도시한 바와 같이 맵(51)상의 윈도우를 소정의 양만큼 모든 주어진 점으로 시프트시킨다. X와 Y 방향으로 맵(51)상의 윈도우(52)의 시프팅 양은  $\Delta X$ 와  $\Delta Y$ 로 나타낸다.

윈도우(52)가 움직임에 따라 CPU(11)는 맵(51)을 표시하는데 새로이 필요한 데이터 블록을 획득한다. 즉, CPU(11)는 데이터 블록(B1-B16)이 필요하다는 것을 검출한다.

하드디스크(21)에서 이들 데이터를 판독하기 위해 CPU(11)는 데이터판독, 블록(B1)의 헤드주소, 제1메모리(24)에 판독데이터의 선(先)기록 등을 지시하는 전송요구신호를 제2버스(27)를 거쳐 전송제어장치(26)와 SCSI 인터페이스(22)에 보낸다. 이 정보에 응답하여 하드디스크(21)는 블록(B1)의 데이터를 순차적으로 판독하고 이 데이터를 SCSI 인터페이스(22)를 거쳐 전송제어장치(26)에 공급한다. 전송제어장치(26)에 있는 제1전송회로(36)는 SCSI 인터페이스(22)에서 하나씩 공급된 맵데이터를 FIFO(37)에 순차적으로 기억시킨다. 제2전송회로(38)는 FIFO(37)에서 데이터를 순차적으로 판독하고 제1메모리(24)의 제1주소카운터(39)에 의해 지정된 위치에 데이터를 기억시킨다.

블록(B1)의 데이터가 하드디스크(21)에서 제1메모리(24)에 전송되는 동안 CPU(11)는 데이터 전송처리로부터 자유롭게 되어 VRAM(14)에 있는 맵데이터의 시프팅 처리를 실행한다.

좀더 자세히 설명하면 VRAM(14)에 있는 윈도우(52)상의 각 비트의 맵데이터는 X주소가  $-\Delta x$ 만큼 시프트되고 Y주소가  $-\Delta y$ 만큼 시프트 되는 위치로 순차적으로 시프트되어 도 7에 도시한 바와 같이 기록된다. 그 결과 윈도우는 도 8a에 있는 상태에서 도 8b에 있는 상태로 시프트된다.

이런 시프팅 처리중에 전송제어장치(26)에서 블록(B1)의 전송종료를 지시하는 인터럽트 신호를 수신할 때 CPU(11)는 인터럽트 처리를 실행하고 다음 블록(B2)을 판독하기 위한 전송요구신호를 명령한다. 이 전송요구신호에 따라 전송제어장치(26) 및 SCSI 인터페이스(22)는 하드디스크(21)에서 블록(B2)의 데이터를 순차적으로 판독하고 이 데이터를 제1메모리(24)에 있는 빈 영역에 기록한다.

VRAM(14)에 있는 윈도우(52)상의 맵데이터의 시프팅이 완료되고 전송종료를 지시하는 인터럽트 신호(INT)가 전송제어장치(26)에서 공급될 때 CPU(11)는 다음 블록(B3)을 판독하기 위한 전송요구신호를 출력한다. 이 전송요구신호에 따라 전송제어장치(26) 및 SCSI 인터페이스(22)는 하드디스크(22)로부터 블록(B3)의 데이터를 순차적으로 판독하고 헤드위치에서 정돈되어 제2메모리(24)에 데이터를 기록한다.

전송제어장치(26)가 데이터를 판독하는 동안 CPU(11)는 제1메모리(24)에 있는 블록(B1)의 맵데이터에서 필요한 부분(윈도우(52)상에 표시되어야 하는 것)을 잘라내고 이것을 VRAM(14)에 있는 연관부분에 기록한다. 마찬가지로 CPU(11)는 표시되는 블록(B2)의 일부를 VRAM(14)에 있는 연관영역에 기록한다.

그런 후에, 다음에 처리되는 블록을 한 메모리에 미리입력시키고 다른 메모리에 입력된 데이터의 필요한 부분을 VRAM(14)에 기록하는 것은 같은 방법으로 병렬로 양자택일로 실행된다. 모든 필요한 맵데이터가 제1 및 제2메모리에 입력될 때 전송동작이 종료되고 반면에 VRAM(14)에 있는 윈도우(52)에 필요한 데이터 기록이 끝날 때 단일 시프트 동작이 완료된다.

위에 기술한 시프팅 처리가 맵(51)에 대한 윈도우(52)의 시프팅이 유지되는 모든 주어진 기간동안 반복적으로 실행된다.

블록(B1 내지 B16)중에서 블록(B11-B16)은 이전 위치에서 윈도우(52)를 표시할 때 사용된 데이터이다. 그러므로 블록(B11-B16)의 데이터가 제1메모리(24)나 제2메모리(25)에 기억될 때 데이터는 하드디스크(21)에서 다시판독되지 않고 메모리에서 남아 있는 데이터를 이용하여 시프팅 처리가 실행될 것이다.

이하 래스터 데이터가 하드디스크에서 판독되고 벡터이미지와 결합되어 새로운 래스터 이미지를 형성하고 이 새로운 래스터 이미지가 하드디스크(21)에 다시 기록되는 경우에 대하여 설명한다.

이 경우에 예를들면 래스터 이미지는 제2메모리(25)에 등록된 후에 제1메모리(24)에 등록된다. 래스터 이미지가 제2메모

리(25)에 등록되는 동안, 백터이미지가 나타나 제1메모리(24)에 미리 기억된 래스터 데이터와 결합된다. 따라서 제2메모리(25)에 있는 래스터 데이터와 새로이 나타난 백터이미지가 결합되는 동안 제1메모리에 있는 결합된 데이터는 하드디스크(21)에 전송되고, 다음에 처리되는 데이터가 하드디스크(21)에서 판독된다.

요컨대 이 실시예에 의하면, CPU(11)가 한 메모리를 액세스하는 동안 다른 메모리와 하드디스크 사이에 데이터가 전송되도록 2개의 메모리가 준비되고, 데이터 처리 및 데이터 전송이 각 메모리에 대해 병행하여 택일적으로 실행된다. 그러므로 CPU(11)는 패스트 메모리를 액세스하는 속도에 가까운 속도로 하드디스크에 있는 데이터를 액세스할 수 있다. 특히 이런 특징은 방대한 양의 데이터를 고속으로 처리할 때 효과적이다.

메모리 수는 정해지지 않는데 2이상의 메모리가 제공될 수 있다. CPU(11)의 속도가 데이터 전송속도에 대하여 느려짐에 따라 메모리 수가 증가되어야 하고, 메모리의 최적수는 각 시스템의 명세에 따라 이용되어야 한다. 그것은 각 메모리의 용량과 맞는다.

비록 위에 기술한 실시예가 고속동작을 위해 개별의 제1버스(12) 및 제2버스(27)를 이용하지만 이들 버스는 도 9에 도시한 바와 같이 공통 버스로 대체될 수 있다.

위에서 설명한 바와 같이 본 발명은 외부기억장치와 CPU 간의 빠른 데이터 전송을 보장하여 빠른 데이터처리를 이룰 수 있다.

## 제2실시예

제1실시예에서, FIFO는 제1 및 제2메모리와 외부기억장치 사이에 제공된다. 타이밍 조정회로를 이용하여 제1 및 제2메모리와 외부기억장치 사이에 데이터가 전송되도록 변경될 수 있다. 이하 타이밍 조정회로를 이용한 제2실시예 대하여 설명한다.

도 10은 본 발명의 데이터 전송장치와 통합되는 컴퓨터 시스템(100)을 도시한 블록도이다. 시스템(100)은 CPU(102)를 연결하는 버스(108), 내부메모리(104), 데이터 전송장치(110)를 포함한다. 본 발명에서 버스(108)는 컴퓨터 시스템에서 전형적으로 발견된 하나 혹은 그 이상의 버스를 참조한다. 버스의 예는 시스템 버스와 주변 버스(예를들면 IBM 호환 컴퓨터 시스템에서 전형적으로 발견된 PCI, 로컬 버스, ISA, EISA 버스)이다. 본 발명에서 내부메모리(104)는 버스(108)를 거쳐 주소를 기입할수 있는 어떤 메모리일 것이다. 예를들면 메모리(104)는 시스템 메모리(전형적으로 시스템버스에 접속됨)이거나 비디오 메모리(전형적으로 주변버스에 접속됨)일 수 있다. 데이터 전송장치(110)는 주변보드로 실행될 수 있거나 반도체 칩에 넣어질 수 있다. 본 발명의 한 실시예에서 메모리(104) 및 CPU(102)는 시스템 버스에 접속되는 반면에 장치(110)는 ISA버스에 접속된 주변보드이다.

또한 컴퓨터 시스템은 데이터를 포함하는 하드디스크(112)를 포함한다. 본 발명의 한 형태는 하드디스크(112)와 메모리(104) 사이에 데이터를 빨리 전송하기 위한 시스템이다. 아래에 설명된 바와 같이, RAM A 및 RAM B로 붙여지고 장치(110) 내에 있는 한쌍의 RAM은 하드디스크(112)와 메모리(104) 사이에 전송된 데이터를 일시적으로 기억하는데 사용된다.

이하 데이터 전송장치(110)의 구조에 대하여 설명한다. 장치(110)는 장치(110)내에 있는 회로를 버스(108)에 접속하기 위한 버스인터페이스(106)를 포함한다. 버스인터페이스(106)가 있어서 한편으론 이들 회로사이의 데이터 및 제어신호가 통신되고 다른 한편으론 메모리(104)와 CPU(102)간에 통신이 이루어지도록 한다.

장치(110)는 소형 컴퓨터 시스템 인터페이스(SCSI) 제어장치(122)를 포함하는데 이것은 데이터 및 제어정보를 표준 SCSI 프로토콜에 있는 하드디스크(112)와 통신하게 한다. 그런 제어장치의 설계는 관련기술분야에서 잘 알려져 있으므로 기술하지 않는다. SCSI 제어장치(122)는 바람직하게 반도체 칩의 형태이다.

SCSI 제어장치(122)는 SCSI 처리회로(124)에 접속된다. SCSI 처리회로(124)는 버스 인터페이스(106) 및 타이밍 조정회로(130)에 접속된다. SCSI 처리회로(124)는 다음의 동작을 실행한다. (1) 버스 인터페이스(106)를 거쳐 CPU(102)로부터 명령 및 상태정보를 수신하고, (2) 이들 명령 및 상태정보에 따라 (SCSI 제어장치(122)를 통해) 하드디스크(112)를 제어하고

, (3) 타이밍 조정회로(130)와 공동으로 RAM 처리회로(132)에서 혹은 처리회로(132)로 데이터를 전송하고, (4) 버스 인터페이스(106)를 거쳐 상태정보를 CPU(102)에 보낸다. 예를들어 SCSI 처리회로(124)는 하드디스크(112)로 혹은 하드디스크(112)로부터 데이터 전송을 활성화 및 비활성화 할 수 있다.

SCSI 처리회로(124)에서 발생된 상태신호는 데이터 전송장치(110)와 함께 동작하도록 설계된 소프트웨어 드라이버에 의해 이용될 것이다. 이하 이 드라이버의 동작을 도 11 및 12와 함께 상세하게 설명한다.

본 발명의 제2실시예에서 하드디스크(112)는 SCSI 표준에 따른다. 그러나 본 발명은 다른 표준, 예를들면 IDE(integrated drive electronics) 혹은 EIDE(enhanced integrated drive electronics)와 부합하는 하드디스크에 동등하게 적용될 수 있다

본 발명에서 데이터는 컴퓨터 시스템(100)의 하드디스크(112)와 내부 RAM(104) 사이에 직접 전송되지 않는다. 대신에 데이터는 데이터 전송장치(110)내에 있는 한쌍의 RAM에 일시적으로 기억된다(RAM A 및 RAM B로 나타냄). RAM A 및 RAM B에 의한 전송은 빠르다. 한편으로 하드디스크(112)에 의한 데이터 전송은 통상적으로 느리고 RAM A 및 RAM B의 데이터 전송과 같지 않다. 타이밍 조정회로(130)는 하드디스크와 RAM A/RAM B 데이터전송을 조화시키고 동기화하는데 사용된다.

위에 기술한 기능을 이룰수 있게 하기 위해 SCSI 처리회로(124)는 상태처리회로(146)(상태정보를 수신하고 처리하고 발생시키기 위함)와 데이터 흐름회로(148)(하드디스크(112)에서 혹은 하드디스크로 데이터 전송을 제어하기 위함)를 포함한다. 이들 회로의 설계는 회로설계분야에서 기술을 가진 자라면 알 수 있을 것이다.

타이밍 조정회로(130)는 RAM 처리회로(132)에 접속된다. 이 RAM처리회로(132)는 버스 인터페이스(106)를 거쳐 CPU(102)와 내부 RAM(104)에 접속된다. 게다가 RAM 처리회로(132)는 데이터 전송장치(110)내에 있는 RAM A 및 RAM B에 접속된다.

RAM 처리회로(132)는 한편으로 타이밍 조정회로(130)에, 다른 한편으로 버스 인터페이스(106)에 RAM A 및 RAM B를 접속시키기 위한 스위치(134)를 포함한다. 특히 SCSI 처리회로(124) 사이에서 데이터를 RAM A나 RAM B중 어느 하나에 전송하는 것이 바람직하다면 스위치(134)는 타이밍 조정회로(130)가 적당한 메모리에 연결되게끔 한다. 유사하게, 내부 RAM(104) 사이에서 데이터를 RAM A나 RAM B중의 어느 하나에 전송하는 것이 바람직하다면 스위치(134)는 버스 인터페이스(106)가 데이터 전송장치(110)에 있는 적당한 메모리에 연결되게끔 한다. RAM A는 주어진 시간에 타이밍 조정회로(130)나 버스 인터페이스(106)의 어느 하나에 접속될 수 있지만 타이밍 조정회로(130)와 버스 인터페이스(106)에 동시에 접속될 수 없다. 유사하게 RAM B는 주어진 시간에 타이밍 조정회로(130)나 버스 인터페이스(106)의 어느 하나에 접속될 수 있지만 이들 둘 모두는 동시에 접속될 수 없다. 또한 데이터는 타이밍 조정회로(130)와 버스 인터페이스(106) 사이에 직접 전송될 수 없다. 이하 데이터 전송과 관련한 스위치(134)의 자세한 동작을 설명한다.

RAM A와 RAM B에 있는 데이터의 양을 감시하기 위하여 RAM 처리회로(132)는 두 바이트 처리장치를 포함한다. 각 장치는 카운터를 포함하고 RAM A나 RAM B와 RAM처리회로(132)사이의 전기신호를 대조하기 위한 수단을 포함한다. 이들 장치는 카운터(136 및 138)를 참조한다. RAM 처리회로(132)는 RAM A와 RAM B로부터 혹은 RAM A와 RAM B로의 데이터 전송과 카운터(136 및 138)의 감소값/증가값을 각각 감시한다.

그 결과 카운터(136 및 138)의 값은 RAM A와 RAM B에 각각 기억된 데이터 양의 지시를 제공한다.

RAM 처리회로(132)의 또다른 기능은 버스 인터페이스(106)를 거쳐 CPU(102)로부터 명령과 상태정보를 수신하는 것이다. RAM 처리회로(132)는 수신된 명령 및 상태정보에 따라 한편으로 내부 RAM(104)과 다른 한편으로 RAM A와 RAM B사이의 데이터 전송을 제어한다. 또한 RAM 처리장치(132)는 상태신호를 CPU(12)에 보낸다. 이들 신호는 데이터 전송장치(110)를 사용하도록 설계되는 소프트웨어 모듈에 의해 이용될 것이다. 상태처리회로(152)는 RAM처리회로(132)에 포함되어 이들 기능을 수행한다.

RAM A와 RAM B의 액세스 시간을 최적화하는 것이 바람직하므로 바람직하게 정적 RAM이 이용된다. 그러나 RAMs의 다른형태(동적 RAM등)가 이용될 수 있어서 본 발명은 RAM의 특정형태로 국한되지 않는다.

한편으로 내부RAM과 다른 한편으로 RAM A 및 RAM B 사이의 데이터 전송은 CPU(102)에 의해 직접(예컨대 CPU에 데이터 전송명령을 내림으로써) 처리될 수 있거나 종래의 직접 메모리 액세스(DMA)에 의해 처리될 수 있다. 출력이 강한 데이터 전송명령이 없는 처리장치에 대해서는 DMA에 의해 전송이 가장 효과적으로 처리된다. 출력이 강한 데이터 전송동작(예를 들면 버스트 모드 전송 및 스트림 전송명령)을 갖는 처리장치에 대해서는 전송은 CPU 자체에 의해 가장 효과적으로 처리된다.

본 발명의 한 실시예에서 SCSI 처리회로(124), RAM 처리회로(132), 타이밍 조정회로(130), 그리고 버스 인터페이스(106)는 단일 반도체 칩(140)상에 제조된다.

하드디스크(112)의 기능 중에 하나는 응용프로그램이나 운영체제에 의해 사용된 데이터 5/또는 프로그램을 기억하는 것이다. 하드디스크(112)와 응용프로그램(혹은 운영체제) 사이에 데이터를 전송할 때 장치(110)의 장점을 충분히 이용하기 위하여, 이들 사이의 동작을 조화시키는데 소프트웨어 드라이버가 바람직하게 이용된다. 소프트웨어 드라이버는 장치(110)에 의해 발생된 상태신호에 응답할 수 있고 명령을 보낼 수 있다.

이하 컴퓨터 시스템(100)의 동작을 설명한다. 제1예에서 응용 소프트웨어는 소프트웨어 드라이버에 명령을 내려 하드디스크(112)에서 어느정도의 데이터의 양을 검색하고 내부메모리(104)에 전송된다. 도 11에 도시한 흐름도(S180)는 이 예와 관련하여 이용된다. 소프트웨어 드라이버는 SCSI 처리회로(124)와 RAM 처리회로(132)에 초기화 명령을 내린다(단계 S182). SCSI 처리회로(124)는 데이터 검색 및 전송을 준비하도록 하드디스크(112)를 (SCSI 제어장치(122)를 통해) 초기화한다. RAM 처리회로(132)는 카운터(136 및 138) 값을 제로로 설정하고 스위치(134)를 설정하여 RAM A에 타이밍 조정회로(130)를 접속시킨다. RAM 처리회로(132)는 SCSI 처리회로(124)에서 데이터를 검색하고(타이밍 조정회로(130)에 의해 처리된 후) 이것을 RAM A에 전송한다. 그 결과로서 데이터가 하드디스크(112)에 판독되고 RAM A에 로드된다(단계 S184).

단계 S184중에 CPU(102)는 연루되지 않으므로 다른 기능을 수행할 수 있다. 즉 윈도우즈나 다른 소프트웨어 모듈을 실행한다.

RAM 처리회로(132)는 카운터(136)를 이용하여 RAM에 있는 데이터의 양이 소정값에 달하는지를 결정한다(단계 S186). 만일 이 값에 달하지 않으면(즉, RAM A가 부가데이터를 수용할 공간을 갖는다면), 타이밍 조정회로(130)에 의해 처리된 후에 SCSI 처리회로(124)(및 하드디스크(112)로부터의 데이터는 RAM A로 계속 흐른다(단계 S188). RAM A가 데이터를 수용할 수 없을 때 RAM 처리회로(132)는 스위치(134)가 RAM B에 타이밍 조정회로(130)를 접속하게끔 한다. 그 결과 데이터가 하드디스크(112)에서 판독되고 RAM B에 로드된다(단계 S190). 이때 RAM 처리회로(132)는 RAM A에 있는 데이터가 메모리(104)에 전송될 수 있도록 소프트웨어 드라이버에 상태신호를 보낸다(단계 S192). 단계 S194에서 소프트웨어 드라이버는 CPU(102)가 사용중인지(즉, 데이터를 전송할 준비가 되어있는지 아닌지)를 결정한다. 만일 CPU(102)가 다른 태스크를 수행하는데 바쁘다면 다른 소프트웨어 드라이버는 기다릴 것이다. 만일 CPU(102)가 데이터 처리에 이용될 수 있다면 소프트웨어 드라이버는 RAM 처리회로(132)에 명령을 보내고 스위치(134)가 RAM A에 버스 인터페이스(106)를 연결하게끔 한다(단계 S196). 이때 데이터는 RAM A에서 내부메모리(104)까지 전송될 수 있다(단계 S198). 위에 설명한 단계 S190(즉, 하드디스크(112)에서 RAM B까지 데이터 전송)은 단계 S192 내지 S198과 동시에 실행될 수 있다는 것을 주목해야 한다. 통상, 단계 S190을 실행하는 시간은 단계 S192-198을 실행하는 시간보다 길다. RAM 처리회로(132)는 단계 S198(즉, RAM A에서 내부 RAM까지 데이터 전송)이 완료될 때 상태신호를 소프트웨어 드라이버에 보낸다(단계 S199). 그 다음에 소프트웨어 드라이버는 다른 태스크를 수행할 수 있다(혹은 다른 소프트웨어 모듈이 다른 태스크를 실행하게끔 한다).

RAM B를 주목하게 되면, RAM 처리회로(132)는 카운터(138)를 이용하여 RAM B에 있는 데이터의 양이 소정값에 달했는지를 결정한다(단계 S200). 만일 이 값에 달하지 않으면(즉, RAM B가 데이터를 수용할 공간이 있다면), 데이터는 하드디스크(112)로부터 RAM B에 계속하여 로드된다(단계 S204). RAM B가 데이터를 수용할 수 없을 때 RAM 처리회로(132)는 스위치(134)로 하여금 RAM A에 타이밍 조정회로(130)를 다시 연결시키게 한다. 흐름도 S180은 경로 S206을 거쳐 단계 S184로 분기하고 RAM A로의 데이터 전송에 관해 위에 기술한 단계가 수행될 것이다. 동시에 RAM 처리회로(132)는 RAM B에 있는 데이터가 메모리(104)에 기억되도록 상태신호를 소프트웨어 드라이버에 보낸다(단계 S210). 단계 S212에서 소프트웨어 드라이버는 CPU(102)가 사용중인지(즉, RAM B에서 데이터를 전송할 수 없는지)를 결정한다. 만일 CPU(102)가 다른 태스크를 실행하는데 바쁘다면 소프트웨어 드라이버는 기다릴 것이다. 만일 CPU(102)가 이용가능하면 소프트웨어 드라이버는 RAM



처리회로(132)에 명령을 보내고 스위치(134)가 RAM B에 버스인터페이스(106)를 접속하게끔 한다(단계 S216). 앞서 지정한 바와 같이, 단계 S210-S216은 하드디스크(112)에서 RAM A까지 데이터의 전송과 함께 동시에 발생할 수 있다. 통상, 하드디스크(112)에서 RAM A까지의 데이터 전송시간은 단계 S210-S216을 실행하는 시간보다 길다. 단계 S216을 완료하자마자 RAM 처리회로(132)는 상태신호를 소프트웨어 드라이버에 보낸다(단계 S217). 소프트웨어 드라이버는 다른 태스크를 수행할 수 있다(혹은 다른 소프트웨어 모듈로 하여금 다른 태스크를 수행하게 한다).

응용 소프트웨어는 (소프트웨어 드라이버를 통해) 하드디스크(112)에서 메모리(104)까지 전송된 데이터의 양을 결정한다. 위에 기술한 사이클은 필요한 모든 데이터가 내부메모리(104)에 전송될때까지 계속된다.

통상, RAM A 및 RAM B에 기억된 데이터를 내부메모리(104)에 전송하는 시간은 디스크 데이터 전송시간에 비할 때 매우 짧다.

따라서 RAM A와 RAM B는 메모리에 기억된 데이터가 내부메모리(104)에 이미 전송되어 있기 때문에 디스크 데이터 전송을 항상 이용할 수 있다. RAM A 및 RAM B 모두가 디스크 데이터 전송을 이용할 수 없다면 소프트웨어 드라이버는 이들 중 하나를 이용할 수 있을 때까지 디스크 전송동작을 지연시켜야 한다.

제2예에서 데이터는 내부메모리(104)에서 검색되고 하드디스크(112)에 기억된다. 도 12에 있는 흐름도 S230은 이 예와 관련하여 이용된다. 이 예에서 응용 소프트웨어는 소프트웨어 드라이버에 명령을 내리고 데이터가 내부메모리(104)에서 검색되고 하드디스크(112)에 전송되도록 한다. 소프트웨어 드라이버는 SCSI 처리회로(124) 및 RAM 처리회로(132)에 초기화 명령을 내린다(단계 S232). 그 다음에 SCSI 처리회로(124)는 데이터를 검색하고 전송할 준비가 되도록 하드디스크(112)를 (SCSI 제어장치(122)를 통해) 초기화 한다. RAM 처리회로(132)는 카운터(136 및 138)의 값을 제로로 설정하고 RAM A에 버스 인터페이스(106)가 접속되게 스위치(134)를 설정한다. 데이터는 내부메모리(104)에서 RAM A까지 전송된다(단계 S234).

카운터(136)는 RAM A에 있는 데이터의 크기가 소정값에 달하는지를 결정하는데 사용된다(단계 S236). 만약 RAM A가 추가 데이터를 수용할 수 있다면 데이터는 내부메모리(104)에서 계속 검색될 것이고 RAM A에 기억된다(단계 S238). RAM A가 추가 데이터를 수용할 수 없을 때 RAM 처리회로(132)는 스위치(134)로 하여금 타이밍 조정회로(130)에 RAM A가 접속되게끔 한다. SCSI 처리회로(124)는 타이밍 조정회로(130)의 제어하에 RAM 처리회로(132)에서 하드디스크(112)까지 데이터를 전송한다(단계 S240). 동시에 RAM 처리회로(132)는 RAM B가 데이터를 기억할 수 있는지를 결정한다(단계 S242). RAM B가 처음으로 액세스 될 때 데이터를 기억할 수 있어야 한다. 그러나, 이것은 RAM A(패스트 반도체-반도체 전송)를 채우기에 앞서, 기억된 데이터를 하드디스크(112)(비교적 느린 처리)에 완전히 전송하기에 충분한 시간이 없을 수도 있기 때문에 나중에 이용할 수 없다. 만일 RAM B를 이용할 수 없다면 RAM 처리회로(132)는 사용불능에 대한 상태신호를 소프트웨어 드라이버에 보낸다(단계 S244). 상태신호를 수신하자마자 소프트웨어 드라이버는 응용 소프트웨어(혹은 운영체제)에 다른 태스크를 실행하라고 알린다(단계 S246).

만일 RAM B를 이용할 수 있다면 RAM 처리회로(132)는 사용가능에 대한 상태신호를 소프트웨어 드라이버에 보낸다(단계 S250). 단계 S252에서 소프트웨어 드라이버는 CPU(102)가 바쁜지(즉, 데이터를 전송할 수 없는지)를 결정한다. 만약에 CPU(102)가 이용될 수 없다면 소프트웨어 드라이버는 기다릴 것이다. 만일 CPU(102)가 이용될 수 있다면 소프트웨어 드라이버는 RAM 처리회로(132)에 명령을 보내고 스위치(134)가 RAM B에 버스 인터페이스(106)를 접속하게끔 한다. 데이터는 내부메모리(104)에서 RAM B까지 전송된다(단계 S254).

카운터(138)는 RAM B에 있는 데이터 크기가 소정값에 달하는지를 결정하는데 사용된다(단계 S256). 그 값에 이르지 못할 때(즉, RAM B는 추가데이터를 수용할 수 있다), 단계 S254가 반복되고 데이터는 내부메모리(104)에서 계속 검색될 것이고 RAM B에 기억된다(단계 S256). RAM B가 추가데이터를 수용할 수 없을 때 RAM 처리회로(132)는 스위치(134)가 타이밍 조정회로(130)에 RAM B를 접속시키도록 한다.

SCSI 처리회로(124)는 타이밍 조정회로(130)의 제어하에 하드디스크(112)에 데이터를 전송한다(단계 S262). 또한 RAM 처리회로(132)는 RAM A가 데이터를 지금 수용할 수 있는지를 결정한다(단계 S264). 만일 RAM A가 이용될 수 없다면(예를 들어, 데이터는 비교적 느린 전송속도 때문에 하드디스크(112)에 아직 기억되지 않았다), RAM 처리회로(132)는 상태신호를 소프트웨어 드라이버에 보낸다. 그리고 이것은 내부메모리(104)에서의 데이터 전송을 일시적으로 중단시킨다(단계 S266).

그 다음에 소프트웨어 드라이버는 응용 소프트웨어(혹은 운영체제)에 통보하고 데이터 전송이 필요하지 않기 때문에 다른 태스크를 지금 실행할 수 있다(단계 S268).

RAM A가 데이터를 수용할 준비가 되지만 하면 RAM 처리회로(132)는 상태신호를 소프트웨어 드라이버에 보낸다(단계 S270). 소프트웨어 드라이버는 CPU(102)가 사용중인지를 결정한다(단계 S272). 사용중이라면(즉, 데이터를 전송할 수 없다면), 소프트웨어 드라이버는 CPU(102)가 사용중일 때까지 기다린다. 만이 CPU(102)가 사용중이 아니라면 RAM 처리회로(132)는 스위치(134)가 RAM A에 버스 인터페이스(106)를 다시 접속하게끔 한다. 이제 데이터는 내부메모리(104)에서 RAM A까지 전송될 수 있다. 흐름도 S230은 단계 S234로 분기하고 전체 알고리즘이 다시 반복된다. 내부메모리에서 RAM A까지의 데이터 전송은 RAM B에서 하드디스크(112)까지의 데이터 전송과 동시에 진행될 수 있다.

응용 프로그램은 (소프트웨어 드라이버를 통해) 메모리(104)에서 하드디스크(112)까지 전송되는 데이터의 양을 결정한다. 위에 설명한 사이클은 필요한 모든 데이터가 하드디스크(112)에 전송될 때까지 계속된다.

본 발명의 한 특징은 하드디스크(112)로부터 및 하드디스크로의 데이터 전송이 CPU(102) 대신에 데이터 전송장치(110)에 의해 처리된다는 것이다. CPU(102)는 단지 고레벨 명령(예컨대 전송처리의 시작 및 종료)을 데이터 전송장치(110)에 내리는 것이다. CPU(102)는 데이터 전송의 세부를 감시하고 처리할 필요가 없다. 그 결과 CPU(102)는 데이터가 하드디스크(112)와 RAM A 혹은 RAM B 사이에 전송되는 동안 다른 태스크를 수행할 수 있다.

도 13은 SCSI 사이클과 도 10에 도시한 컴퓨터 시스템의 내부메모리 사이클 사이의 관계 타이밍을 도시한다.

타이밍 도(280)는 하드디스크(112)와 RAM A/RAM B 사이의 데이터 전송을 도시한다. 도 13에서 참조부호(282 및 283)는 하드디스크(112)와 RAM A 사이에 데이터를 전송하는 시간간격에 대응하는 반면, 참조부호(286 및 287)는 하드디스크(112)와 RAM B 사이에 데이터를 전송하는 시간간격에 대응한다. 참조부호(285)로 지정되고 RAM A와 RAM B 사이에 스위칭하는 시간간격은 데이터를 전송하는 시간간격보다 훨씬 짧다. 이것은 스위칭이 RAM 처리회로(132)에 있는 반도체 소자를 전기적으로 이용하여 실행되기 때문이다.

타이밍 도(270)는 내부메모리(104)와 RAM A/RAM B 사이의 데이터 전송을 도시한다. 도 13에서 참조부호(272 및 273)는 내부메모리(104)와 RAM A 사이에 데이터를 전송하는 시간간격에 대응하는 반면, 참조부호(278 및 277)는 내부메모리(104)와 RAM B 사이에 데이터를 전송하는 시간간격에 대응한다. 시간간격(272,273,276,277)(내부메모리 전송용)은 시간간격(282,283,286,287)(하드디스크 전송용)보다 훨씬 짧다.

위에 지정한 바와 같이, 컴퓨터 시스템(100)은 하드디스크 전송용 시간간격(282,283,286,287)동안 다른 기능(예컨대 운영체제나 응용프로그램을 실행)을 수행할 수 있다. 그 결과 컴퓨터 시스템(100)의 처리작업량은 종래 시스템보다 훨씬 크다.

RAM A 및 RAM B의 크기는 성능과 비용의 기능이다. RAM A와 RAM B를 위한 126킬로바이트의 메모리 크기는 합리적인 비용으로 적절한 성능을 가져온다는 것을 알았다. RAM A와 RAM B는 다른 크기를 가질 수 있다는 것을 알아둘 필요가 있다. 그래서 RAM A와 RAM B의 전체크기는 이용될 필요가 없다. 예를들어 도 11의 단계 S186 및 단계 S200과 도 12의 단계 S236 및 S256에서 소정값은 RAM A와 RAM B의 전체크기(예컨대 128킬로바이트)보다 작을 것이다.

RAM A와 RAM B는 캐시메모리가 아니라는 것을 알아둘 필요가 있다. 따라서 히트(hits) 및 미스(misses)를 검출하고 히트빈도수에 따라 메모리 일부를 대체하기 위한 회로를 포함할 필요가 없다. 본 발명의 회로는 간단하고 빠르다.

디스크 전송을 빠르게 하기위해 버퍼를 이용하는 종래기술의 시스템이 있다. 예를들면 많은 하드디스크는 처리량을 향상시키기 위해 버퍼로서 반도체 메모리를 이용한다. 그러나 이들 시스템은 RAM A/RAM B 아키텍처 및 연관 소프트웨어 드라이버를 사용하지 않는다. 따라서 이들의 성능은 본 발명의 시스템 성능보다 열등하다.

본 발명의 애플리케이션중에 하나는 마이크로소프트(MS)윈도우즈에 기초한 소프트웨어 애플리케이션으로 사용하기 위해 하드디스크에 기억된 그래픽 파일을 고속으로 검색하는 것이다. 현대의 많은 컴퓨터 설계에서 표시메모리(비디오 RAM, 즉

VRAM등)는 아날로그 신호로 변환되는 정보의 디지털 이미지를 기억하는데 이용되고 표시장치(컴퓨터 모니터등)상에 표시된다. MS 윈도우즈 하에서 애플리케이션은 컴퓨터 화면상에 윈도우로 나타날 수 있다. 윈도우의 크기는 모니터의 전체크기에서 모니터 표시영역의 소부분까지의 범위에 있다. 따라서 표시메모리의 모든 혹은 소부분은 디지털 이미지를 기억하는데 이용될 수 있다.

본 발명을 예시하는데 사용된 그래픽 파일은 수 메가바이트의 그래픽 데이터를 포함하는 큰 파일이다. 이것은 하드디스크에 기억된다. 한편으로 통상의 컴퓨터 시스템에서 VRAM의 크기는 약 1메가바이트이다. 만일 그래픽 이미지를 포함하는 표시 윈도우가 작은 윈도우라면 디지털 이미지로 된 VRAM의 양은 1메가바이트보다 훨씬 작다. 두 경우에서 하드디스크에 있는 그래픽파일은 VRAM에 있는 디지털 파일의 크기보다 훨씬 크다.

윈도우에 있는 이미지를 화면이동시키는 것이 바람직하다면 하드디스크에 기억된 그래픽파일의 데이터는 표시메모리에 다운로드되어야 한다. 화면이동 속도는 그래픽 데이터의 다운로드 속도에 의해 종종 제한된다. 따라서 하드디스크와 표시메모리 사이에서 데이터를 빨리 전송하기 위한 시스템을 갖는 것이 바람직하다.

위의 여러 실시예에서 시스템 RAM은 시스템 RAM의 이용가능 크기가(통상적으로 MS 윈도우즈를 실행할 수 있는 컴퓨터에서 적어도 4메가바이트) VRAM의 크기보다 크기 때문에 중간기억영역으로 이용된다. 하드디스크에서 검색되고 시스템 RAM에 로드된 그래픽 데이터의 양은 VRAM의 크기를 초과할 수 있다. 그래픽 데이터의 일부가 표시윈도우에 표시되는 그래픽 데이터와 약간만 다르게 표시되어야 한다면 그래픽 데이터는 시스템 RAM에서 VRAM까지 전송될 수 있다. 그 결과 이 상황에서 디스크 데이터 전송을 시작할 필요가 없다. 시스템 RAM과 VRAM 사이의 그래픽 데이터 전송은 RAM-RAM 전송용 하드웨어 및 소프트웨어에서 오버헤드가 매우 낮기 때문에 매우 빠르다. 따라서 이런 변용은 어느정도까지 처리량을 증가시킬 수 있다. 그러나 만일 화면이동속도가 빨라 시스템 RAM에 아직 로드되지 않는 그래픽 데이터가 자주 필요하다면 디스크 데이터 전송은 피할 수 없다.

IBM 호환 퍼스널 컴퓨터에서 하드디스크 동작은 장치 서비스 루틴(DSR)에 기초한 ROM에 의해 제공된다. 이 루틴은 기본 입출력 시스템(BIOS)소프트웨어의 일부인데 이것은 운영체제, 장치 드라이버, 및 셸로 구성된다. 하드디스크 DSR은 INT(13) 기능을 호출함으로써 액세스된다. DSR은 공업표준 하드디스크 제어장치와 통신하고 제어한다. 이것으로 응용프로그램은 판독, 기록, 초기화, 테스트, 포맷 등을 포함하는 하드디스크에 대한 많은 동작을 수행할 수 있다. 또한 상태정보(예를들면 에러코드)를 응용프로그램에 리턴시킨다.

MS 윈도우즈의 몇가지 버전에서 이들 동작을 포함하는 많은 소프트웨어 애플리케이션은 하드디스크와 시스템 RAM 사이에 데이터를 전송하기 위하여 INT(13)를 사용한다. 그러나 이 방법은 데이터 전송속도가 매우느리다. 그 이유중에 하나는 MS 윈도우즈 코드를 실행할 때 CPU가 많은 사이클을 소비할 필요가 있다는 것이다. 따라서 하드디스크를 최대용량으로 구동시킬 수 없다. 그 결과로서 데이터 전송중에 하드디스크는 일시 정지해야 되고 CPU 사이클을 기다려야 한다. 또다른 이유는 INT(13) 및 종래 하드디스크 인터페이스가 데이터 전송을 처리하는 많은 CPU 사이클을 필요로 하여 기본동작을 위해 귀중한 CPU 자원을 이용한다는 것이다. 그 결과 응용 프로그램은 CPU 자원에 대해 BIOS와 경쟁해야 하기 때문에 천천히 실행될 것이다. 하드디스크와 시스템 RAM(혹은 VRAM) 간의 데이터 전송을 향상시키기 위하여 본 발명의 데이터 전송장치가 이 시스템에 유리하게 사용될 수 있다.

도 14는 MS 윈도우즈 환경에서 본 발명을 이용하여 그래픽 데이터를 전송하도록 설계된 컴퓨터 시스템(340)의 개략도를 도시한다. 컴퓨터 시스템(340)은 머더보드(도시되지 않음), 모니터(도시되지 않음), 키보드(도시되지 않음)를 구비하는 퍼스컬 컴퓨터(356)를 포함한다. 또한 컴퓨터 시스템(340)은 운영체제(컴퓨터의 BIOS와 공동으로 작업하는 마이크로소프트사의 MS-도스등) 위에 실행되는 MS 윈도우즈(342)를 포함한다. 운영체제(354)는 퍼스컬 컴퓨터(356)의 자원을 감시하고 제어한다. MS 윈도우즈는 파선(344-347)으로 도 14에 개략적으로 도시한 많은 프로그램 모듈과 절차호출의 층을 포함하는 복잡한 소프트웨어이다.

시스템(340)은 종래 하드디스크 인터페이스(364)에 접속된 종래 하드디스크(360)를 포함한다. 이 인터페이스는 퍼스컬 컴퓨터(356)에서 주변버스(예를들면 ISA 혹은 PCI 버스)에 삽입된 인쇄배선 회로기판일 수 있다. 이것은 퍼스널 컴퓨터(356)가 하드디스크(360)와 통신하고 제어하도록 하는 전자부품(하드디스크 제어장치 칩등)을 포함한다. 통상 하드디스크



라 불리는 종래의 소프트웨어 프로그램(366)은 하드디스크(360)와 하드디스크 제어장치의 조직과 구조를 알지 않고도 응용프로그램이 하드디스크(360)를 제어하고 액세스할 수 있도록 한다. 위에 지정한 바와 같이 하드디스크 드라이버는 컴퓨터(356)의 BIOS에 포함될 것이다. 그러나 운영체제에서 개별적으로 실행하는 개별 하드디스크 드라이버 소프트웨어를 갖는 것이 가능하다.

애플리케이션 380 및 381과 같은 애플리케이션은 MS 윈도우즈(342) 상부에서 실행된다. 이 예에서 애플리케이션 380과 같은 애플리케이션중의 하나는 본 발명을 예시하는데 이용된다. 애플리케이션(380)은 MS 윈도우즈(342)와 운영체제(354)에서 긴 체인의 프로그램 모듈을 통해 하드디스크 드라이버(366)와 통신한다. 따라서 도 15에서 HD 드라이버(366)는 파선(344-347) 아래에 위치한 부품과 운영체제(354)의 일부로서 개략적으로 도시된다. 위에 지정한 바와 같이 종래기술의 하드디스크 드라이버는 하드디스크로부터 혹은 하드디스크로의 데이터 전송시에 효과적이지 못하다. 게다가 애플리케이션(380)과 하드디스크 드라이버(366) 사이의 통신은 MS윈도우즈 모듈과 운영체제(354)의 적어도 일부를 통해야 한다. 그 결과 데이터 전송이 매우 느리다.

본 발명에 따른 디스크 처리시스템은 하드디스크(370), 데이터전송 인터페이스(372), 데이터전송(소프트웨어) 드라이버(374)를 포함한다. 데이터 전송 인터페이스(372)는 도 10의 데이터 전송장치(110)에 대응한다. 본 발명에서, 한편으로 데이터 전송드라이버(374)와 다른 한편으로 MS 윈도우즈(342) 및 운영체제(354)사이의 대화(interaction)를 최소화하는 것이 바람직하다. 이런 배치는 더 적은 오버헤드가 필요하므로 실행속도를 향상시킨다는 것을 알아냈다. 도 14에서 데이터 전송 드라이버(374)는 파선(344) 만의 아래에 위치하여 개략적으로 도시된다. 따라서 애플리케이션(380)은 긴 체인의 MS 윈도우즈 모듈을 통하지 않고 데이터 전송 드라이버(374)를 액세스할 수 있다.

본 발명에서 데이터전송 드라이버(374)는 FGB.DLL 및 MDC.DRV로 표제된 2개의 프로그램 모듈을 포함한다. MDC.DRV는 하드디스크(370)를 직접 제어하는 드라이버 프로그램이다. FGB.DLL은 애플리케이션(380)과 MDC.DRV 사이에 인터페이스를 제공하는 윈도우즈 동적 연결 라이브러리이다. 이 시스템에서 MDC.DRV는 종래 하드디스크 드라이버 및 운영체제(354)에 포함된 연관 디스크관련 루틴을 대체하고, MDC.DRV는 도 11 및 12에 있는 흐름도에 따라 데이터 전송 인터페이스(372)와 대화한다.

도 15는 본 발명에 따라 애플리케이션(380)과 하드디스크(370)의 대화를 도시한 흐름도(S400)이다. 이런 도식적 예에서도 11 및 12에 도시한 소정값은 RAM A와 RAM B의 전체크기(즉, 128킬로바이트)로 설정되었다고 가정하자. 또한 도식적 의도로 그래픽 데이터는 시스템 RAM 같은 중간기억매체를 이용하는 대신에 VRAM에 직접 기록된다고 가정하자. 애플리케이션(380)은 MS 윈도우즈하의 표시윈도우 크기와 하드디스크(370)에 있는 연관 그래픽 데이터 파일의 위치를 결정한다(예컨대, 디렉토리/파일명). 애플리케이션(380)은 FGB.DLL을 호출하고 적당한 매개변수를 보낸다(단계 S404). 매개변수의 예는 디렉토리/파일명, 원하는 그래픽 데이터 크기, 윈도우의 위치등이다. 그 다음에 FGB.DLL은 MDC.DRV를 호출한다(단계 S406). 위에 지정한 바와 같이, MDC.DRV는 운영체제 호출을 이용하지 않고 데이터 전송 인터페이스(372)를 직접 제어한다.

MDC.DRV는 데이터전송 인터페이스(372)에 명령을 보내 하드디스크(370)로부터 헤더정보를 요구한다(단계 S408). 헤더정보는 통상 하드디스크(370)의 소정위치에 위치하고 디렉토리명, 파일명, 각 파일의 크기, 각 파일의 시작주소 등을 포함한다. 헤더정보는 MDC.DRV를 거쳐 FGB.DLL에 전달된다(단계 S412).

헤더정보를 수신하자마자 FGB.DLL은 VRAM에 보내야 하는 그래픽 데이터의 위치 및 크기를 결정한다. 단계 S416에서 FGB.DLL은 MDC.DRV에 명령을 보내 하드디스크(370)에 있는 지정된 위치에서 필요한 그래픽 데이터의 양을 요구한다. RAM A와 RAM B의 크기가 128킬로바이트이므로 MDC.DRV는 데이터전송 인터페이스(372)를 제어하여 하드디스크(370)에서 128킬로바이트의 데이터를 판독한다(단계 S418). MDC.DRV는 많은 CPU 사이클을 갖는 동작 때문에 하드디스크(370)에서 데이터를 검색하는 중에 다른 태스크를 수행하라고 FGB.DLL이나 MS윈도우즈에 알린다(단계 S420). 그 결과, 데이터 검색은 다른 애플리케이션의 다른 MS 윈도우즈 모듈과 동시에 실행될 수 있다.

데이터전송 인터페이스(372)는 RAM A나 RAM B가 채워지는지를 (카운터를 통해) 계속하여 감시한다(단계 S422). 만일 하드디스크가 128킬로바이트의 데이터를 검색하지 않았다면 CPU 자원은 다른 소프트웨어 프로그램에 의해 계속하여 사용될 수

있다. 128킬로바이트의 데이터가 검색될 때 데이터전송 인터페이스(372)는 MDC.DRV에 알린다(단계 S424). MDC.DRV는 하드 디스크(370)에서 더 많은 데이터가 검색되어야 하는지를 결정한다(단계 S426). 만일 데이터가 더 필요하다면 MDC.DRV는 데이터 전송 인터페이스(370)에 명령을 보내 하드디스크(370)에서 또다른 128킬로바이트의 데이터를 검색하라고 요구하고 RAM A/RAM B중의 다른 하나에 전송한다(단계 S428). MDC.DRV는 미리 검색된 128킬로바이트의 데이터(RAM A 혹은 RAM B에 기억됨)를 FGB.DLL에 보내고(단계 S430), VRAM에 데이터를 차례로 전송한다(단계 S432). 흐름도(S400)는 단계 S420으로 분기된다(즉, 다른 소프트웨어에 알리고 실행되게 한다).

하드디스크(370)에서 추가데이터를 검색할 필요가 없다면, MDC.DRV는 미리 검색된 128킬로바이트의 데이터(RAM A 혹은 RAM B에 기억됨)를 FGB.DLL에 보내고(단계 S436), 그 데이터를 VRAM에 차례로 전송한다(단계 S438). FGB.DLL은 애플리케이션(380)으로 되돌아간다(단계 S440).

FGB.DLL과 MDC.DRV 사이의 분담은 위의 흐름도에 설명한 것에서 다를 수 있다. 위의 예에서 FGB.DLL은 MDC.DRV에 필요한 데이터 양을 알리고 필요한 양이 검색될 때 MDC.DRV가 데이터 전송을 종료하라는 결정을 하도록 한다. 다른 실시예에서, 이 동작은 FGB.DLL에 의해 실행되는 것이 바람직하다. 즉 FGB.DLL은 데이터 전송 종료를 결정할 수 있다. 이들과 기타 변용은 프로그래밍에 기술을 가진자에 의해 쉽게 만들어질 것이기에 더 이상 기술하지 않는다.

또한 MDC.DRV는 VRAM이나 시스템 RAM에서 하드디스크(370)까지 데이터 전송을 처리할 수 있다. 이 상황에 대한 흐름도는 도 15의 흐름도(S400)와 도 12의 흐름도(S230)에 도시한 정보를 이용하여 기술을 가진자에 의해 쉽게 그려질 수 있다. 따라서 여기서 이들 흐름도는 도시하지 않는다.

데이터전송 인터페이스(372)(하드웨어) 및 연관 데이터전송 드라이버(374)(소프트웨어)는 공동작용하여 하드디스크(370)와 VRAM(혹은 시스템 RAM) 간의 데이터전송의 처리량을 증가시킨다. 하드웨어는 처리량을 증가시킬 수 있다. 그러나 소프트웨어 드라이버가 있어서 다른 애플리케이션 및 운영체제는 컴퓨터의 자원을 완전히 이용할 수 있다.

본 발명은 특정예의 실시예를 참고로 설명되었다. 본 발명의 의도나 범위를 벗어나지 않고 다양한 변용 및 변경이 이루어질 수 있다. 예를들어 본 발명은 애플사의 컴퓨터 및 마이크로소프트사의 윈도우즈의 다른 버전(예컨대 버전3.1, 3.11, NT 및 윈도우즈 95) 등의 현재 이용할 수 있고 미래에 개발되는 다양한 컴퓨터에 이용될 수 있다. 따라서 명세서 및 도면은 제한적 의미, 즉 제공된 청구범위만으로 제한되기 보다는 예증으로 간주되어야 한다.

#### (57) 청구의 범위

청구항 1. 외부기억장치와;

제1 및 제2메모리(24,25)와;

상기 제1 및 제2메모리를 양자택일로 액세스하여 데이터를 처리하는 데이터 처리수단(11)과;

상기 제1 및 제2메모리와 상기 외부기억장치에 접속되며, 상기 데이터 처리수단에서의 명령에 따라, 상기 데이터 처리수단이 액세스하지 않는 상기 제1 및 제2메모리중의 하나와 상기 외부기억장치 사이에 데이터를 전송하기 위한 데이터 전송수단(11)을 포함하는 것을 특징으로 하는 데이터 처리장치.

청구항 2. 제1항에 있어서, 상기 데이터 처리수단이 상기 외부기억장치에서의 데이터 판독을 명령할 때, 상기 데이터 전송수단은 상기 데이터 처리수단에서의 명령에 따라, 상기 외부기억장치로부터 상기 데이터 처리수단에 의해 처리되는 데이터를 판독하고 상기 데이터 처리수단이 액세스하지 않는 상기 제1 및 제2메모리 중의 하나에 상기 데이터를 미리기억시키는 것을 특징으로 하는 데이터 처리장치.

청구항 3. 제1항에 있어서, 상기 데이터 처리수단이 상기 외부기억장치로의 데이터 기록을 명령할 때, 상기 데이터

전송수단은 상기 데이터 처리수단에서의 명령에 따라, 상기 데이터 처리수단이 액세스하지 않는 상기 제1 및 제2메모리중의 하나로부터 처리된 데이터를 판독하고 상기 처리된 데이터를 상기 외부기억장치로 전송하고 데이터가 상기 데이터 처리수단에 의해 기록될 수 있는 영역을 미리형성하는 것을 특징으로 하는 데이터 처리장치.

청구항 4. 제1항 내지 제3항중 어느 한 항에 있어서, 상기 데이터 전송수단은:

데이터를 외부기억장치와 교환하기 위한 제1전송수단(36)과;

데이터를 상기 제1 및 제2메모리와 교환하기 위한 제2전송수단(38)과;

제1 및 제2전송수단 사이에서 데이터 전송을 수행하기 위한 버퍼메모리(37)를 포함하는 것을 특징으로 하는 데이터 처리장치.

청구항 5. 제1항 내지 제3항 중 어느 한 항에 있어서, 상기 데이터 전송수단은:

데이터를 상기 외부기억장치(112)와 교환하기 위한 제1전송수단(124, 148)과;

데이터를 상기 제1 및 제2메모리(RAM A, RAM B)와 교환하기 위한 제2전송수단과;

상기 제1 및 제2전송수단 사이의 전송속도차를 조정하기 위한 타이밍 조정회로(130)를 포함하는 것을 특징으로 하는 데이터 처리장치.

청구항 6. 외부기억장치에 접속되는 포트(31)와;

제1 및 제2메모리(24, 25)와;

상기 제1 및 제2메모리를 양자택일로 액세스하여 데이터를 처리하는 데이터 처리수단에 접속되는 포트(32)와;

상기 제1 및 제2메모리와 상기 외부기억장치에 접속되며, 상기 데이터 처리수단에서의 명령에 따라 상기 데이터 처리수단이 액세스하지 않는 상기 제1 및 제2메모리중의 하나와 상기 외부기억장치 사이에서 데이터를 전송하기 위한 데이터 전송수단(26)을 포함하는 것을 특징으로 하는 인터페이스 회로.

청구항 7. CPU(102), 외부기억장치(112), 및 내부메모리(104)를 구비하는 컴퓨터 시스템에서 데이터를 전송하기 위한 시스템에 있어서:

제1기억장치(RAM A)와;

제2기억장치(RAM B)와;

상기 제1기억장치에 기억된 데이터의 양을 결정하기 위한 상기 제1기억장치에 연결된 제1카운터(136)와;

상기 제2기억장치에 기억된 데이터의 양을 결정하기 위한 상기 제2기억장치에 연결된 제2카운터(138)와;

상기 제1기억장치에 기억된 데이터의 양이 제1의 소정값을 초과할 때 제1신호를 발생시키기 위한 수단(152)과;

상기 제2기억장치에 기억된 데이터의 양이 제2의 소정값을 초과할 때 제2신호를 발생시키기 위한 수단(152)과;

데이터를 상기 외부기억장치와 통신하기 위한 제1회로와;

데이터를 상기 내부메모리와 통신하기 위한 제2회로와;

상기 제1신호에 응답하여 상기 제1 및 제2회로중의 하나에 상기 제1기억장치를 양자택일로 연결시키고 상기 제2신호에 응답하여 상기 제1 및 제2회로중의 하나에 상기 제2기억장치를 양자택일로 연결시키기 위한 데이터 스위칭회로(134)를 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 8. 제7항에 있어서, 상기 외부기억장치는 하드디스크이고 상기 제1회로는 상기 하드디스크의 동작을 제어하기 위한 제어장치(122)를 추가로 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 9. 제8항에 있어서, 상기 제1회로와 상기 데이터 스위칭수단 사이에 연결되고 상기 하드디스크와 상기 제1 및 제2기억장치 사이의 전송속도차를 조정하기 위한 타이밍 조정회로(130)를 추가로 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 10. 제7항에 있어서, 상기 제2회로는 상기 내부메모리와 상기 제1 및 제2기억장치 사이에 상기 CPU가 데이터를 전송할 수 있게 하기 위한 수단을 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 11. 제10항에 있어서, 상기 수단은 제어신호를 발생시키기 위한 수단을 포함하며,

상기 시스템은 상기 내부메모리와 상기 제1 및 제2기억장치 사이에 데이터를 전송하기 위한 상기 제어신호에 응답하는 소프트웨어 루틴을 추가로 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 12. 제7항에 있어서, 상기 제1 및 제2기억장치는 정적 RAM인 것을 특징으로 하는 데이터 전송시스템.

청구항 13. CPU(102)와 실행되는 적어도 하나의 응용프로그램을 갖는 컴퓨터에서, 외부기억장치(112)와 내부메모리(104) 사이에 데이터를 전송하기 위한 시스템에 있어서:

제1기억장치(RAM A)와;

제2기억장치(RAM B)와;

데이터를 상기 외부기억장치와 통신하기 위한 제1회로와;

상기 제1 및 제2기억장치와 상기 제1회로에서 활동에 응답하여 상태신호를 발생시키기 위한 상태신호회로(152)와;

상기 제1회로와 상기 내부메모리중의 하나에 상기 제1기억장치를 양자택일로 연결시키고 상기 제1회로와 상기 내부메모리중의 하나에 상기 제2기억장치를 양자택일로 연결시키기 위한 데이터 스위칭회로(134)와;

데이터가 상기 제1 및 제2기억장치중의 하나와 상기 외부기억장치 사이에 전송되는 일부시간 동안에 상기 CPU가 상기 적어도 한 프로그램을 실행하게 하기 위한 상기 상태신호중의 제1에 응답하는 소프트웨어 루틴을 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 14. 제13항에 있어서, 상기 소프트웨어 루틴은 데이터가 상기 외부기억장치와 상기 제2기억장치 사이에 전송되는 일부시간 동안에 데이터가 상기 내부메모리와 상기 제1기억장치에 전송되게 하기 위한 상기 상태신호중의 제2에 응답하는 것을 특징으로 하는 데이터 전송시스템.

청구항 15. 제14항에 있어서, 상기 소프트웨어는 상기 CPU가 상기 내부메모리와 제1메모리 사이에 데이터를 전송할 수 있는지를 결정하고, 상기 CPU를 이용할 수 있을 때 상기 전송이 이루지게 하기 위한 수단을 추가로 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 16. 제13항에 있어서, 상기 제1기억장치에 기억된 데이터 양을 결정하기 위한 상기 제1기억장치에 연결되는 제1카운터(136)와 상기 제2기억장치에 기억된 데이터 양을 결정하기 위한 상기 제2기억장치에 연결된 제2카운터(138)를 추가로 포함하고,

상기 상태신호회로는 상기 제1기억장치에 기억된 데이터의 양이 제1의 소정값을 초과할 때와 상기 제2기억장치에 기억된 데이터의 양이 제2의 소정값을 초과할 때에 상태신호를 발생시키는 것을 특징으로 하는 데이터 전송시스템.

청구항 17. 제13항에 있어서, 상기 외부기억장치는 하드디스크이며 상기 제1회로는 상기 하드디스크의 동작을 제어하기 위한 제어장치(122)를 추가로 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 18. 제17항에 있어서, 상기 제1회로와 상기 데이터 스위칭회로 사이에 연결되고, 상기 하드디스크와 상기 제1 및 제2기억장치 사이의 전송속도차를 조정하기 위한 타이밍 조정회로(130)를 추가로 포함하는 것을 특징으로 하는 데이터 전송시스템.

청구항 19. 실행되는 적어도 하나의 응용프로그램, CPU, 제1기억장치, 및 제2기억장치를 포함하는 컴퓨터 시스템에서 외부기억장치와 내부메모리 사이에 데이터를 전송하는 방법에 있어서:

상기 제1기억장치와 상기 외부기억장치 사이의 제1스위칭가능 데이터 전송경로와, 상기 제2기억장치와 상기 외부기억장치 사이의 제2스위칭가능 데이터 전송경로를 제공하는 단계와;

상기 제1데이터 전송경로를 이용하여 상기 제1기억장치에 상기 외부기억장치를 접속시키는 단계와;

상기 제1기억장치에 있는 데이터의 양이 제1의 소정값을 초과할 때 상기 외부기억장치와 상기 제2기억장치 사이에 데이터가 전송될 수 있도록 상기 제2데이터 전송경로를 스위칭하는 단계와;

상기 제2기억장치에 있는 데이터의 양이 제2의 소정값을 초과할 때 상기 외부기억장치와 상기 제1기억장치 사이에 데이터가 전송될 수 있도록 상기 제1의 데이터 전송경로를 스위칭하는 단계와;

상기 제1 및 제2기억장치중의 하나와 상기 외부기억장치 사이에 데이터가 전송되는 일부시간 동안에 상기 응용프로그램을 실행하는 단계를 포함하는 것을 특징으로 하는 데이터 전송방법.

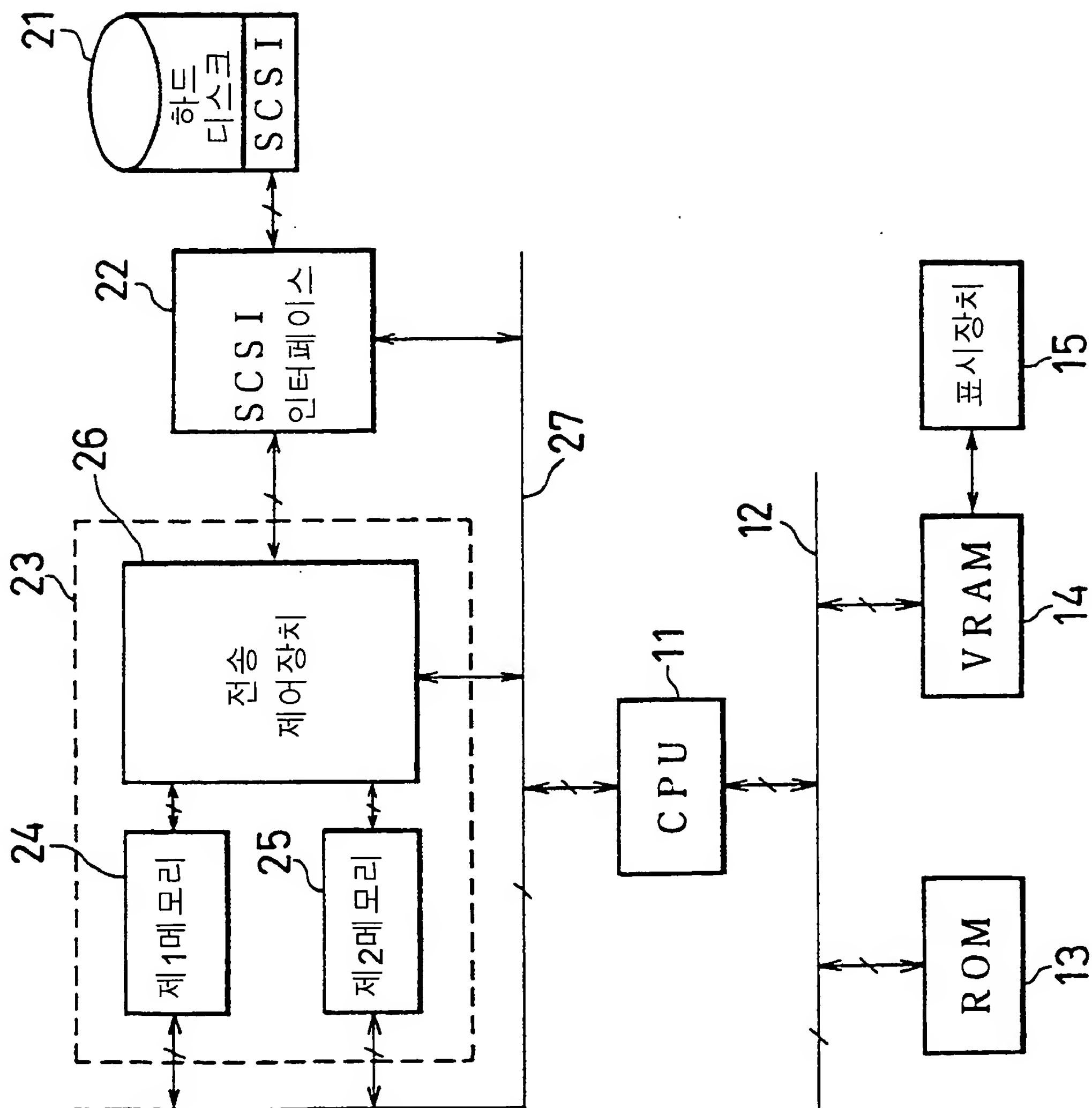
청구항 20. 제19항에 있어서, 상기 제1기억장치와 상기 내부기억장치 사이의 제3스위칭가능 데이터 전송경로와, 상기 제2기억장치와 상기 내부기억장치 사이의 제4스위칭가능 데이터 전송경로를 제공하는 단계와;

상기 외부기억장치가 상기 제2데이터 전송경로를 거쳐 상기 제2기억장치에 접속된 후에만 상기 제3데이터 전송경로를 이용하여 상기 내부기억장치와 상기 제1기억장치 사이에 데이터가 전송되도록 하는 단계와;

상기 외부기억장치가 상기 제1데이터 전송경로를 거쳐 상기 제1기억장치에 접속된 후에만 상기 제4의 데이터 전송경로를 이용하여 상기 내부기억장치와 상기 제2기억장치 사이에 데이터가 전송되도록 하는 단계를 추가로 포함하는 것을 특징으로 하는 데이터 전송방법.

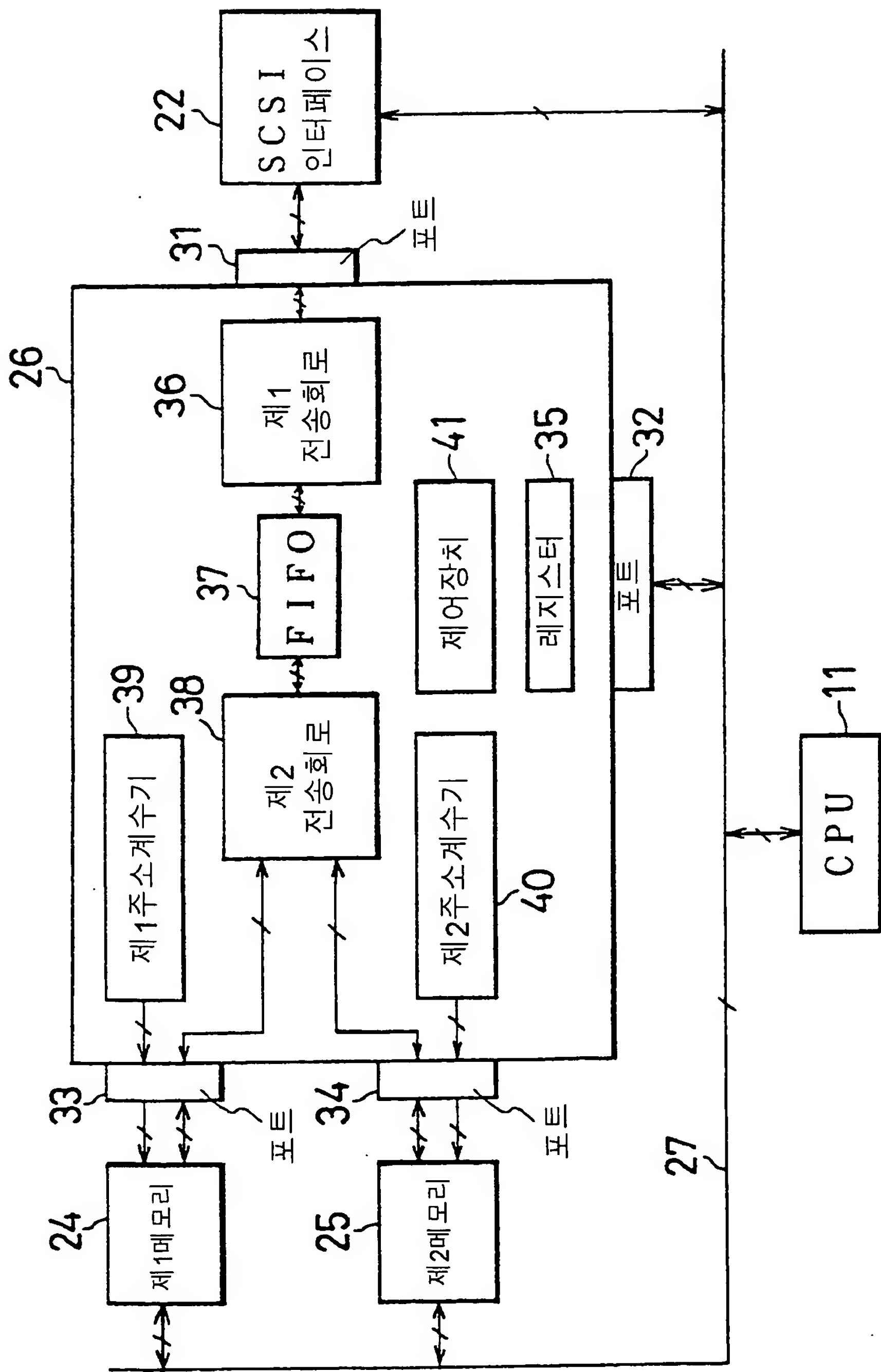
청구항 21. 제19항에 있어서, 상기 제1 및 제2기억장치중의 하나와 상기 내부기억장치 사이에 데이터가 전송되도록 하는 상기 단계는 상기 제1 및 제2기억장치중의 하나와 상기 외부기억장치 사이에 데이터가 전송되는데 필요한 일부시간 동안에 일어나는 것을 특징으로 하는 데이터 전송방법.

도면 1

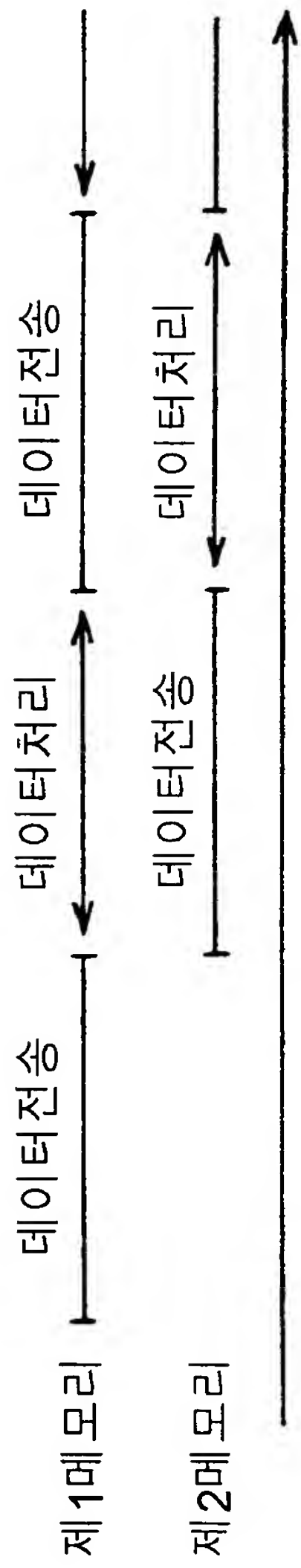


도면 2

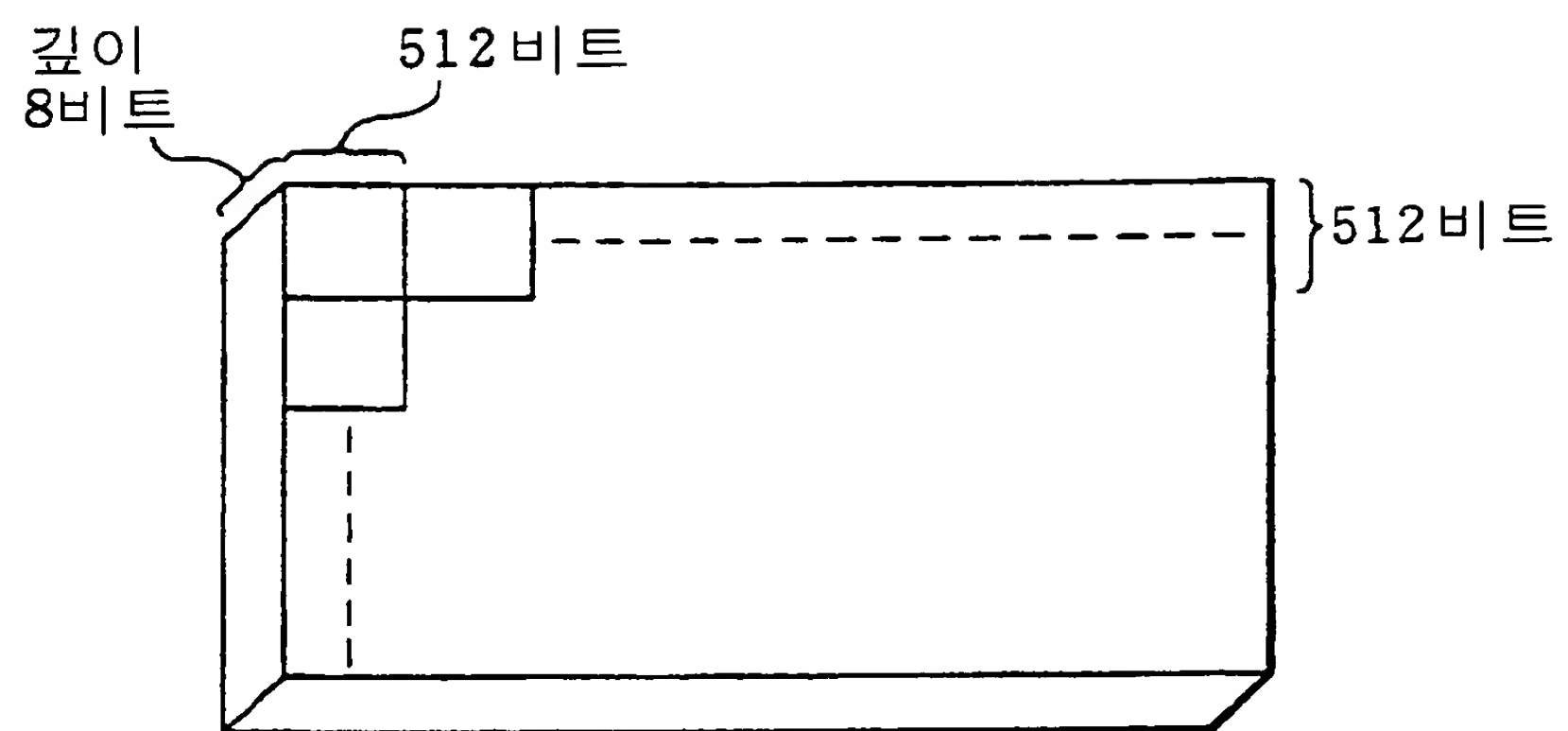
도면3



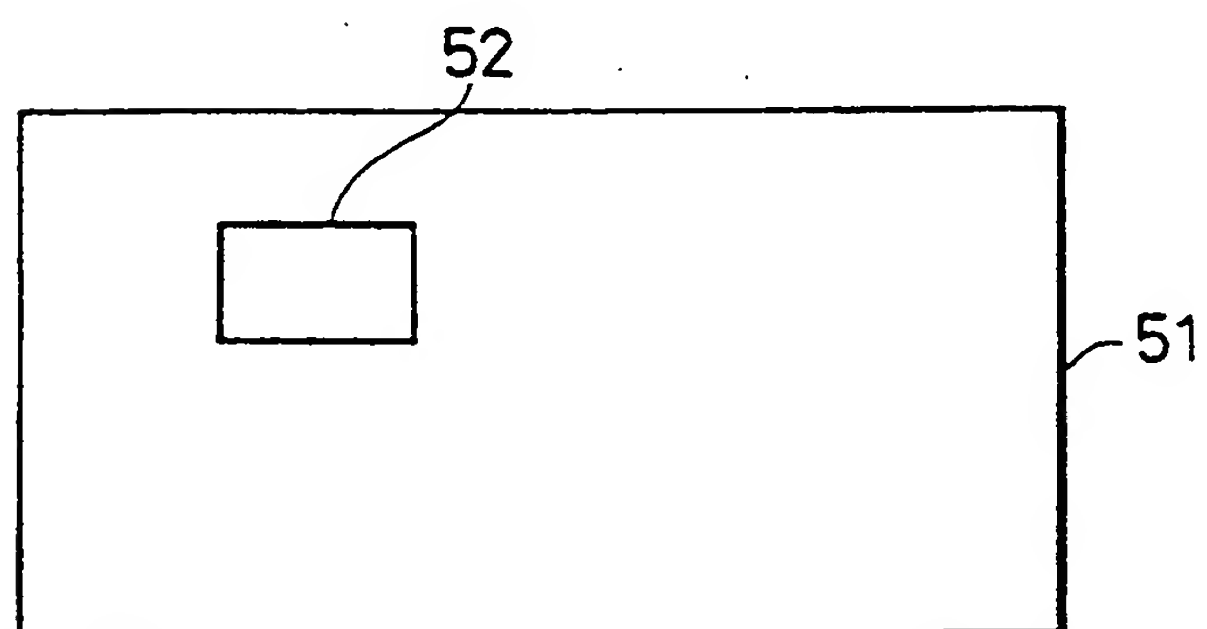




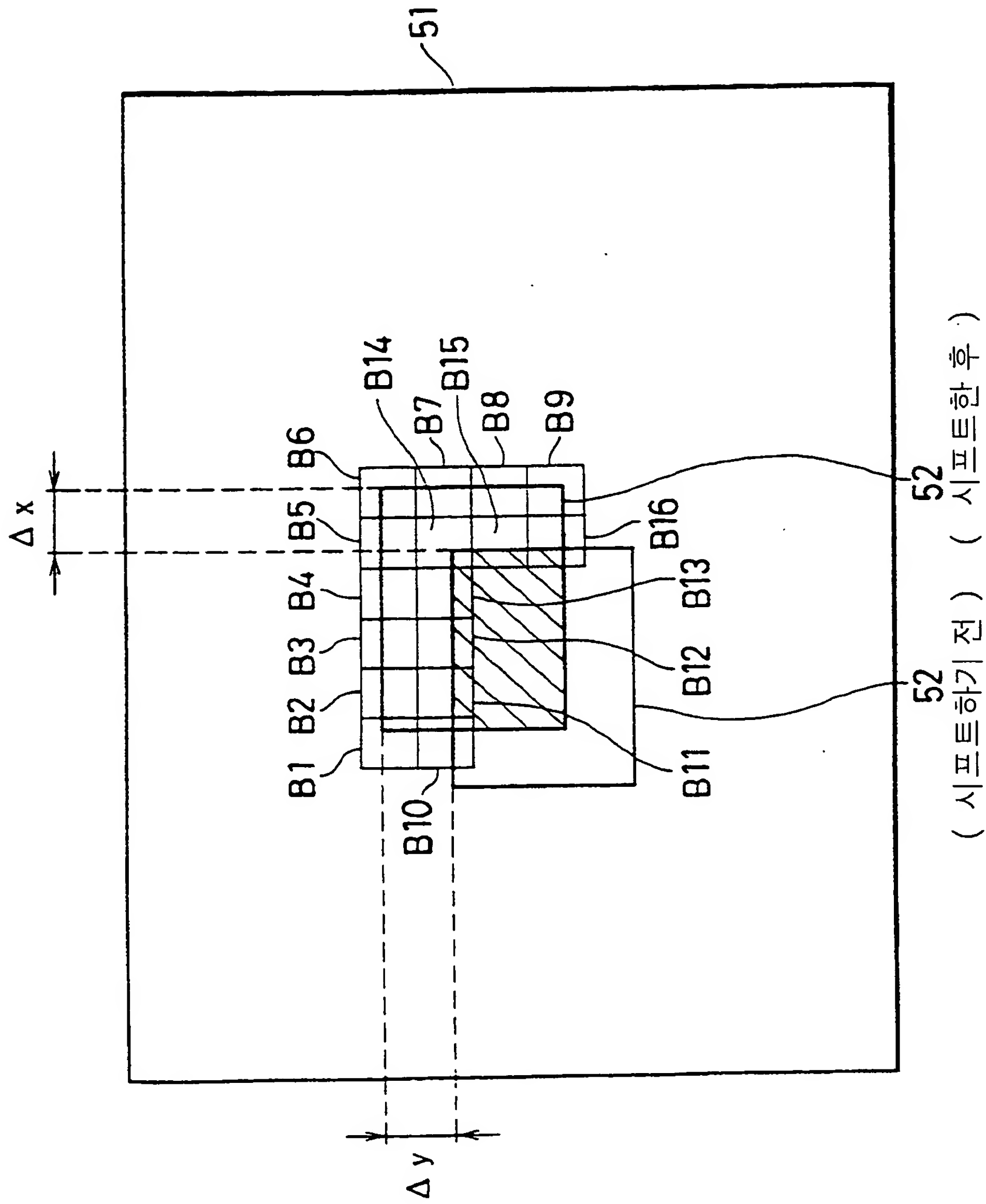
도면4

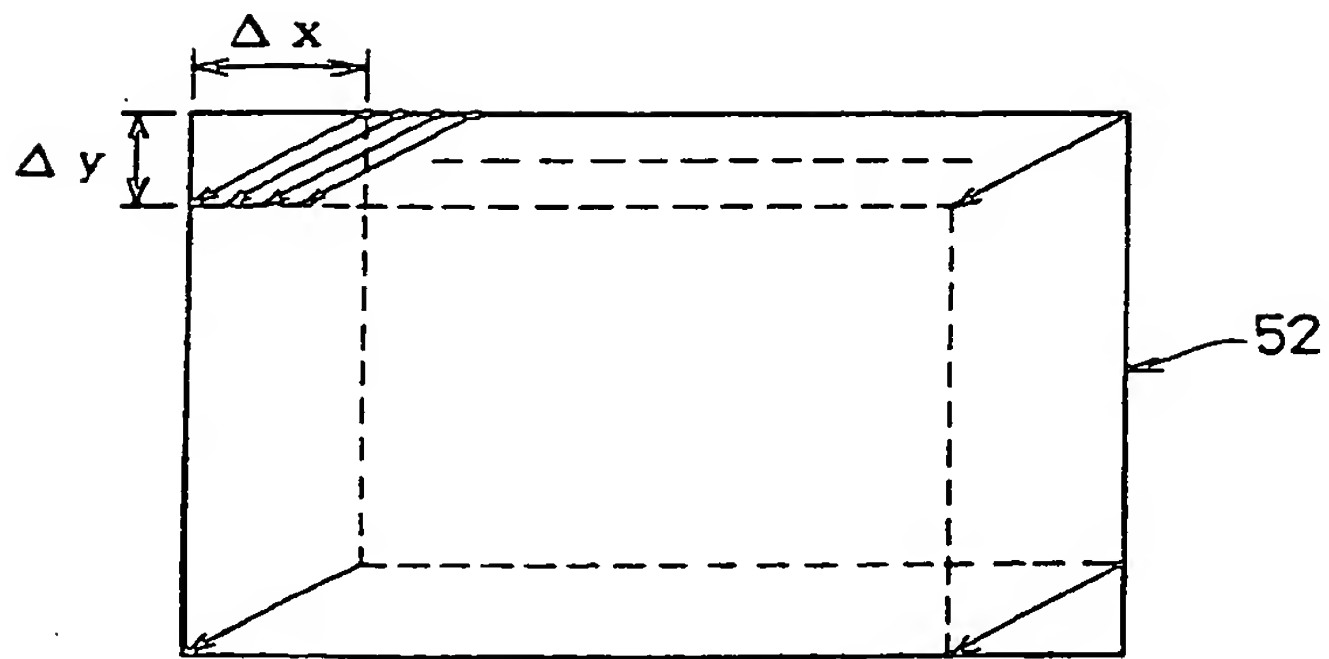


도면5

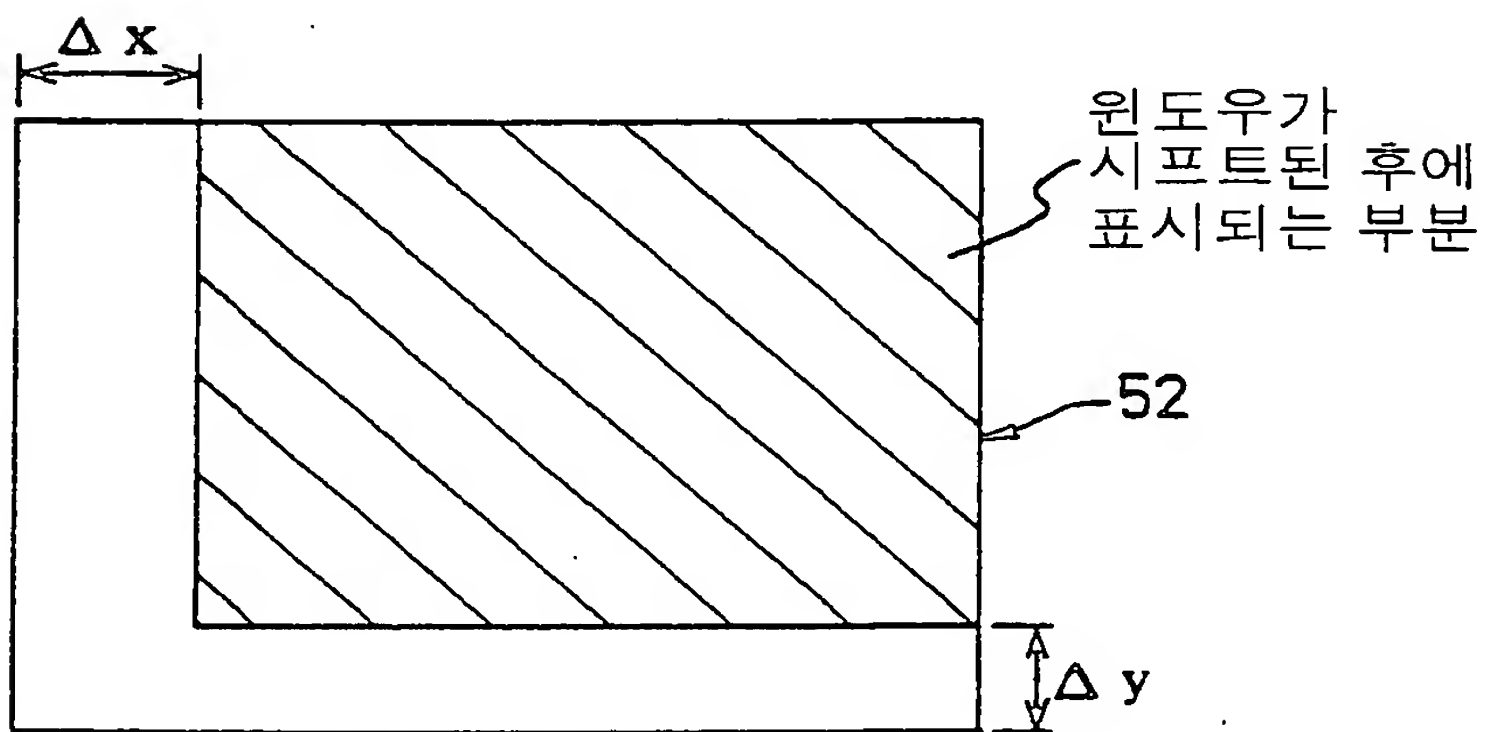


도면6

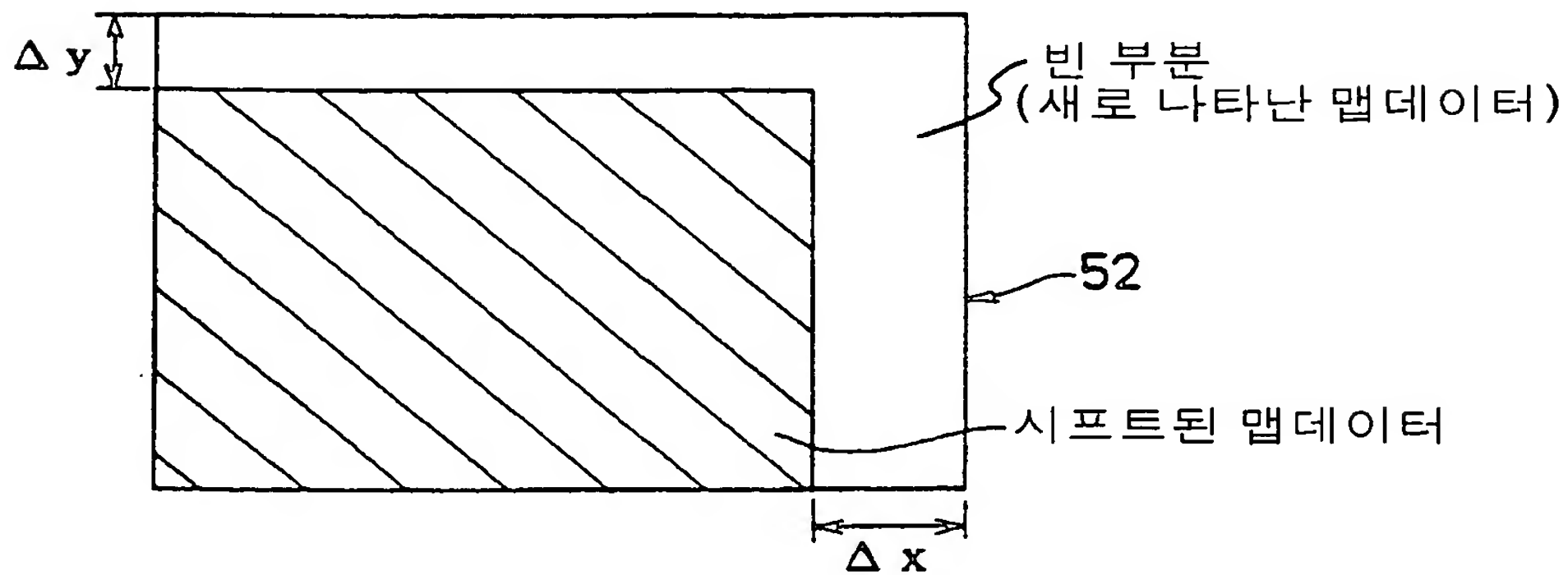




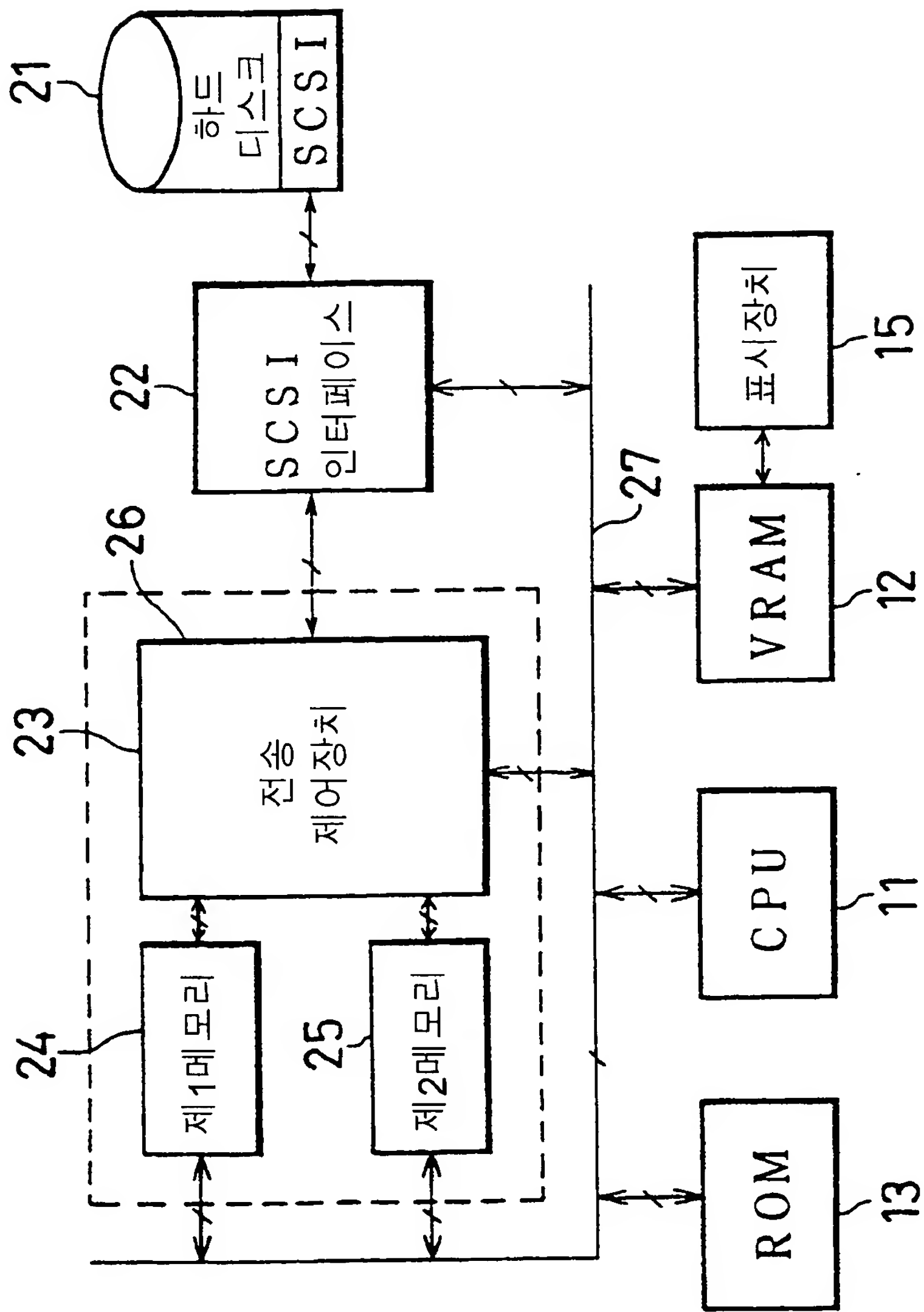
도면 8a



도면 8b

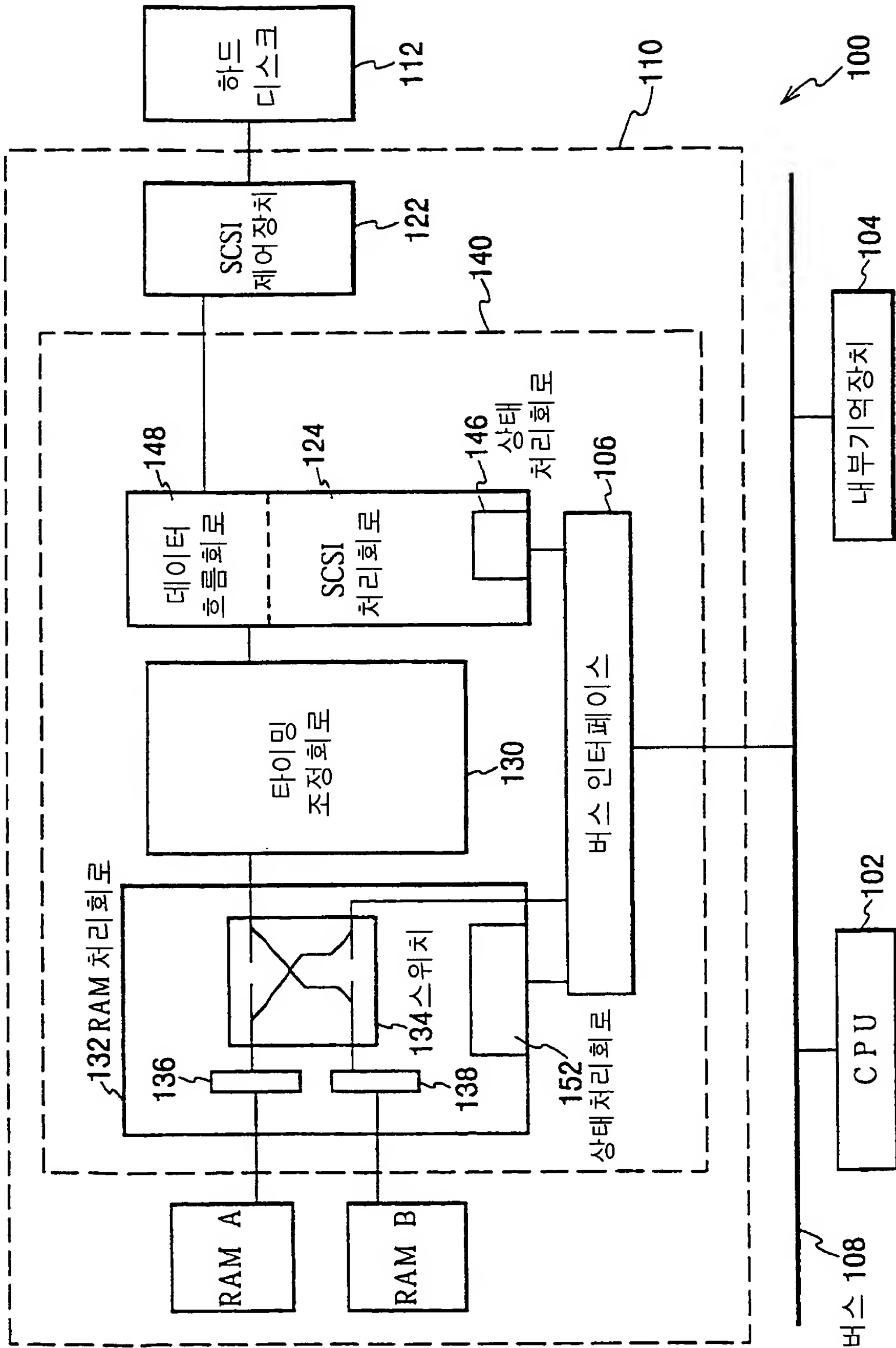


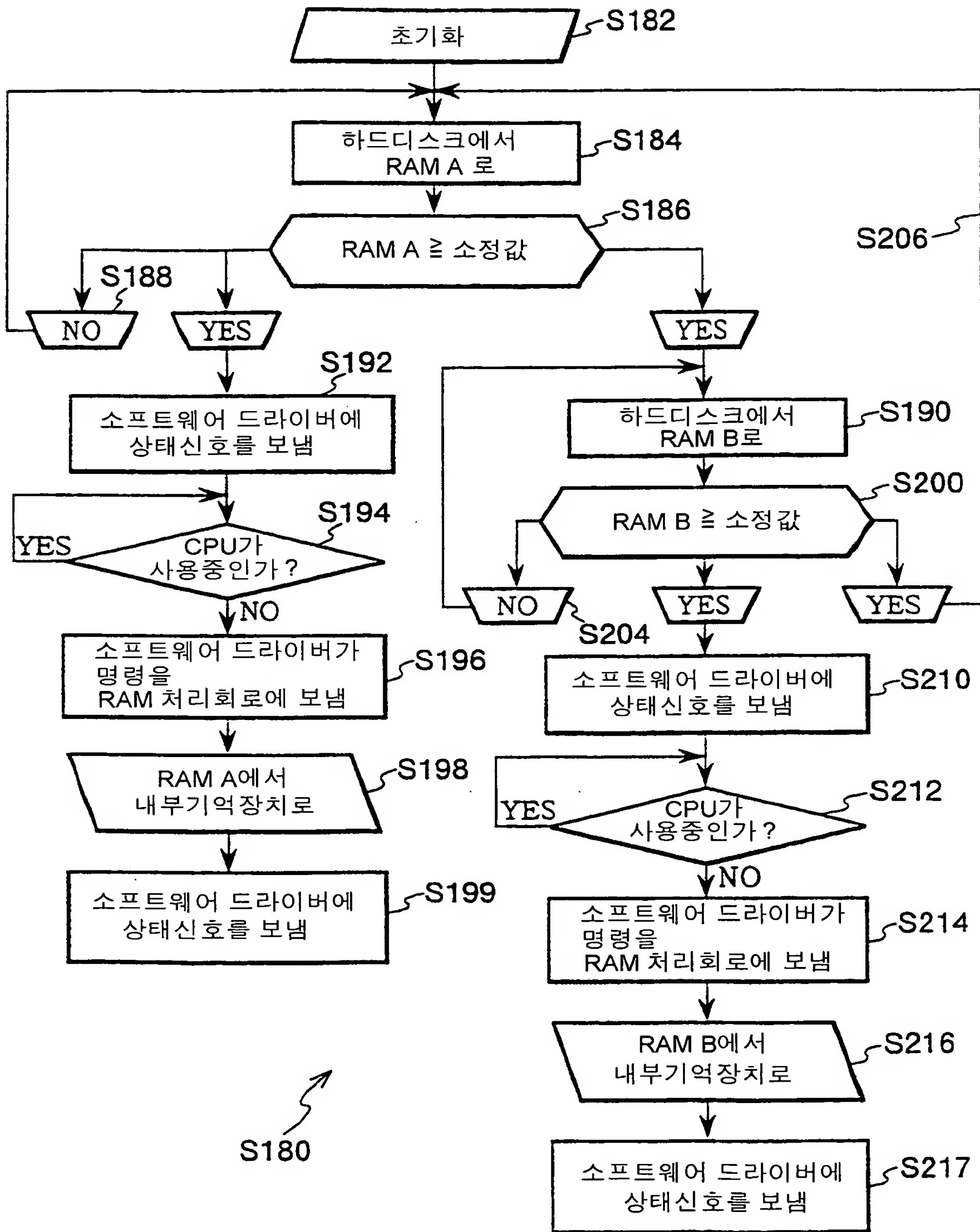
도면9



도면10

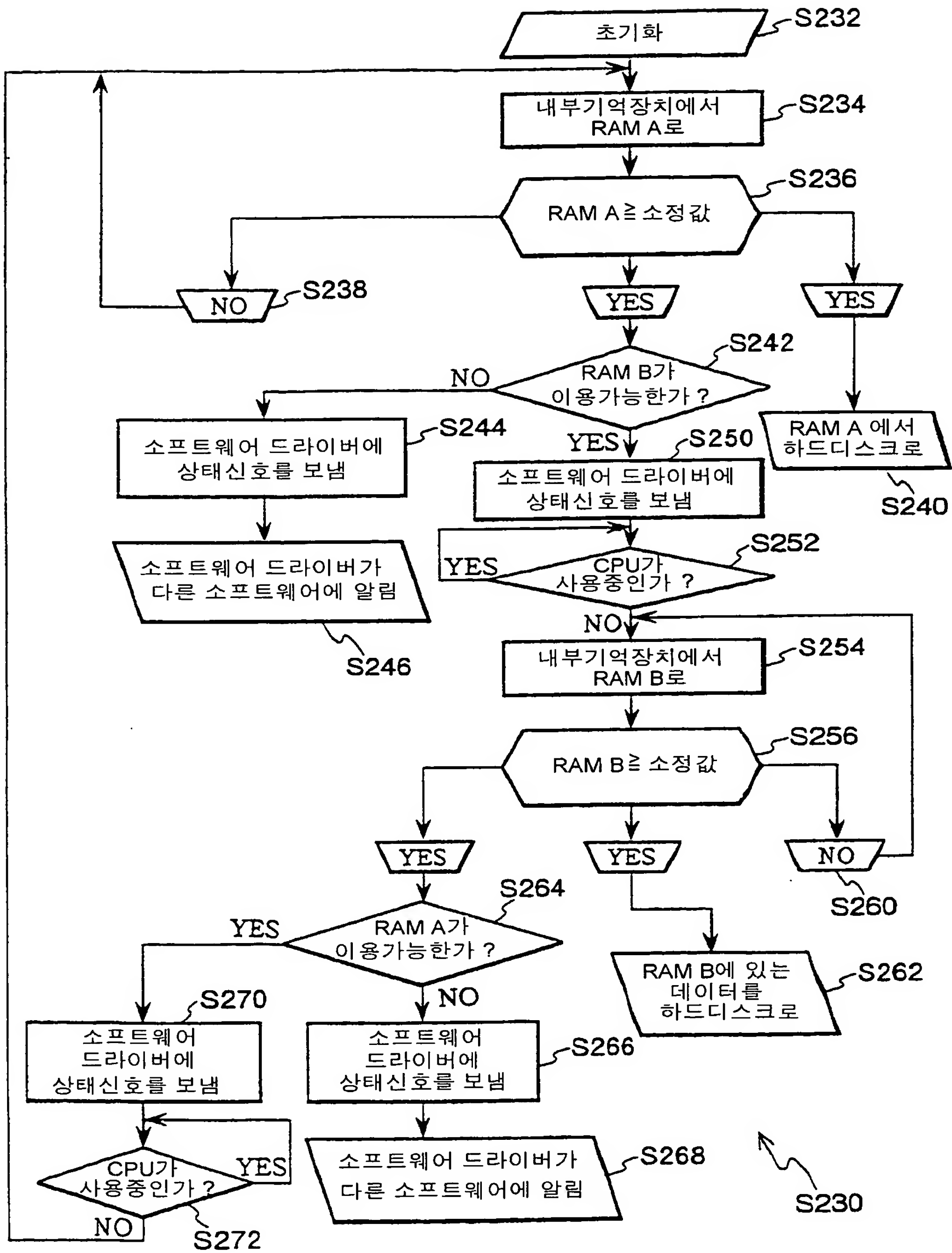
도면 11



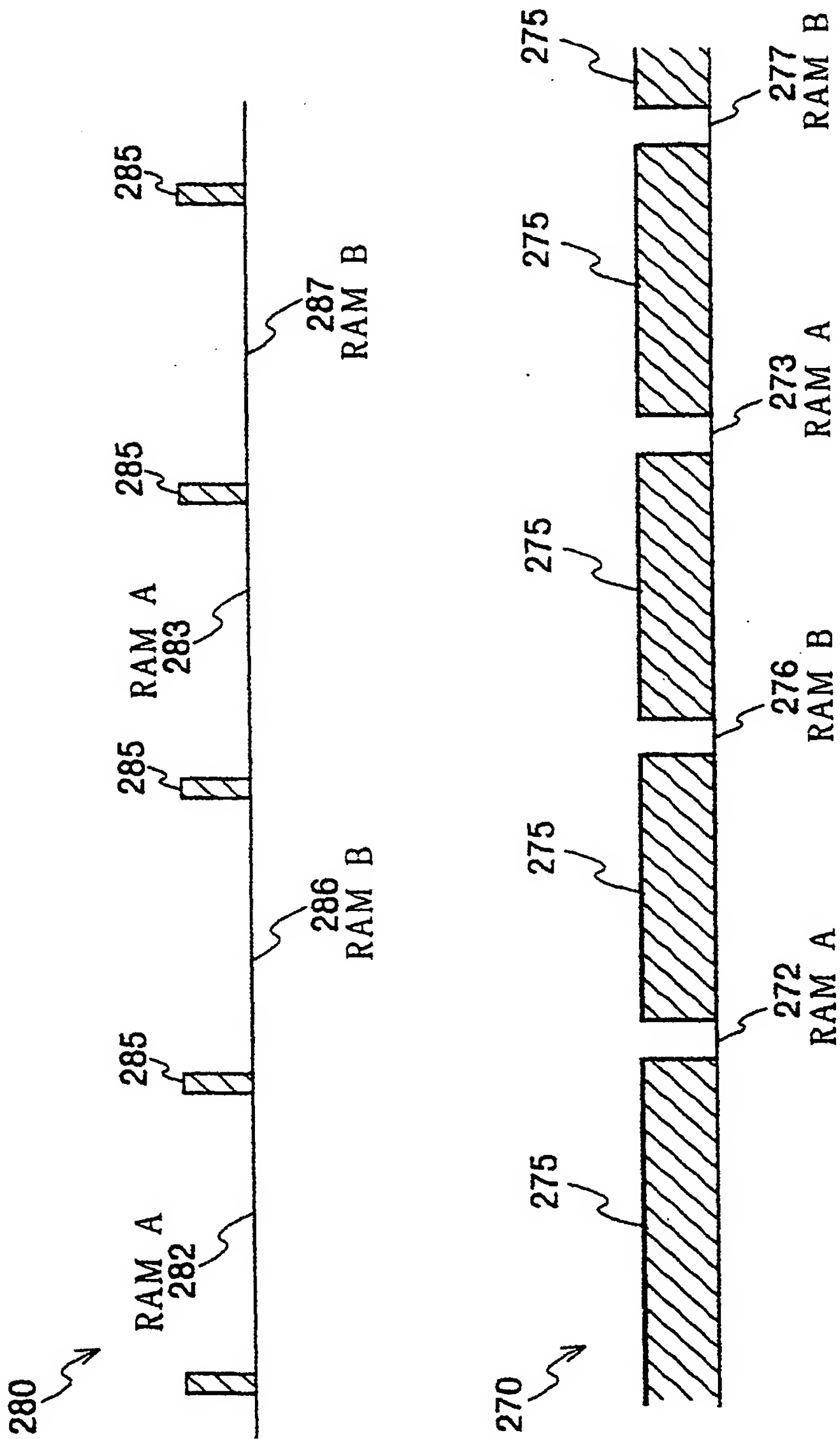


도면 12





도면 13



도면 15

